Detection of COVID-19 and cardiovascular diseases By ECG

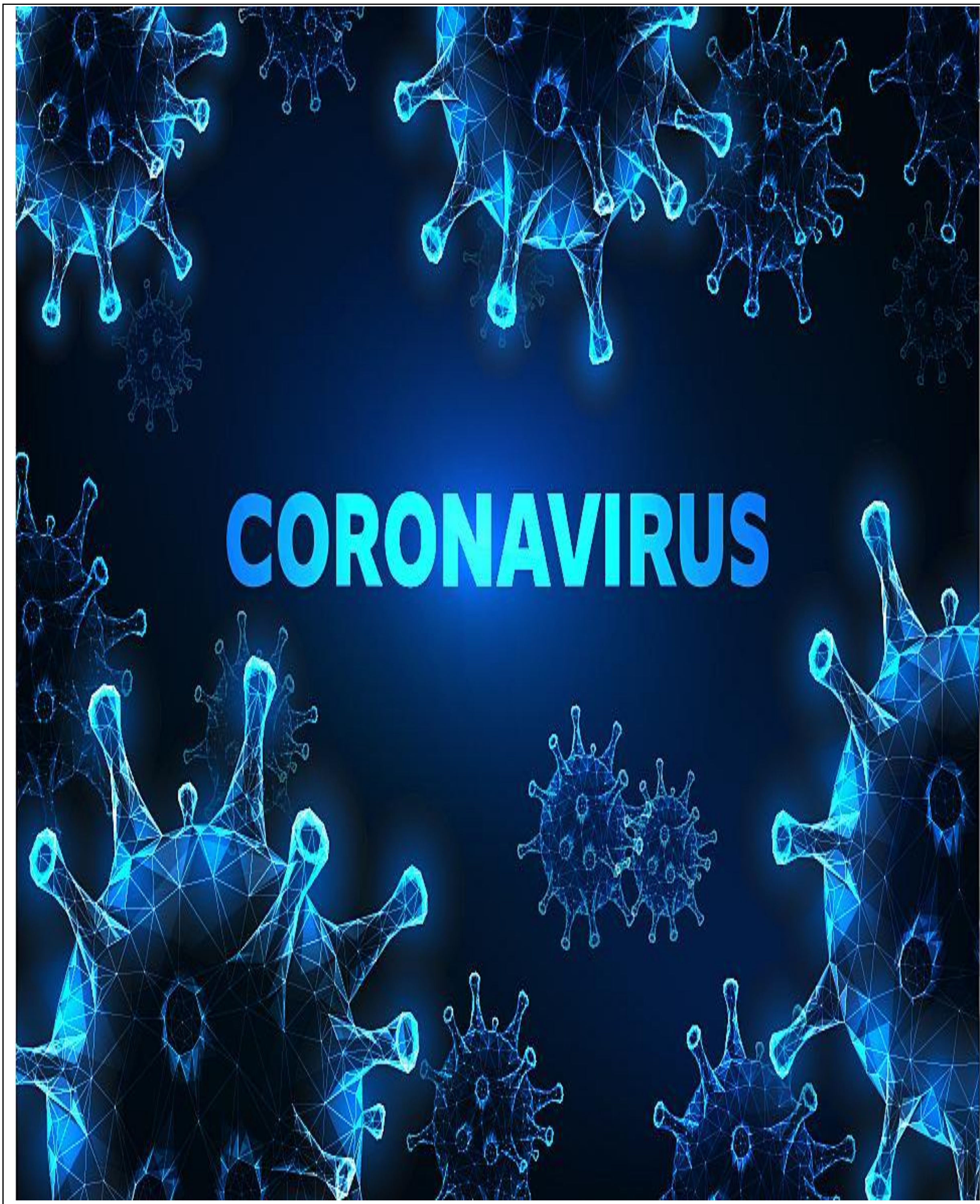
Prepared By:

- David Hanna Khwaga 20-00267
- Jonathan Ehab Nemr 20-00512
- Hager Hamed AbdElrad 20-00217
- Aya Eid Mahmoud 20-00966
- Semon Mamdouh Beskales 20-01833
- Huda Gamal Ahmed 20-00282
- Yara Salman Abdelhalem 20-01414

Under Supervision:

Dr. Mahmoud Bassiouni
Professor in Computers and Information Technology
At Egyptian E-Learning University (EELU)

T.A.Hayam Abdelbaset
Teacher Assistant in Computers and Information Technology
At Egyptian E-Learning University (EELU)


CORONAVIRUS

Acknowledgement:

First, we would like to thank Allah for helping us to complete this project successfully.

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

Thanks to Egyptian E-Learning University especially Suhag center for helping us to reach this level of awareness.

We would like to express our deep and sincere gratitude to our project supervisor ***DR.Mahmoud Bassiouni*** for giving us the opportunity to lead us and providing guidance throughout this project.

We are also especially indebted to expressing our thanks to ***T.A.Hayam Abdelbaset*** for his constant guidance and supervision.

Finally, we would like to express our gratitude to everyone who helped us during the graduation project.

Abstract:

The advent of artificial intelligence has led to AI better investigation of many complex research problems from a variety of domains.

Recently deep learning approaches have emerged as cutting-edge AI technologies and has been proved very effective in medical research.

Many studies have recently used deep learning in the field of medical imaging for the detection and identification of various diseases including COVID 19.

In this work, we have investigated an ECG based approach for detection of COVID-19 and heart diseases using deep learning.

Table of content

List of Figure:

List of tables

List of Acronyms or Abbreviations

. CNN: Convolutional Neural Network

. VGG-19: Visual Geometry Group-19

. ResNet-50: Residual Network-50

. AI: Artificial Intelli GAP: Global Average Pooling

. **ECG**: Electrocardiogram.

. **rembg**: Remove Background.gence

. **GAP** : Global Average Pooling

Chapter 01
Introduction

1.1 history of covid19

The COVID-19 pandemic, caused by the novel coronavirus SARS-CoV-2, is one of the most significant global health crises of the 21st century. The first known cases of COVID-19 were reported in Wuhan, China, in **December 2019*. The disease quickly spread, leading to an outbreak that would affect millions worldwide.

Origins:

The initial cases in Wuhan were linked to the Huanan Seafood Wholesale Market, and it was suspected that the virus might have jumped from animals to humans. The market was shut down on January 1, 2020. The exact origins of the virus are still debated, with theories suggesting a natural spillover from wildlife or a possible accidental release from a virology lab in Wuhan. U.S. intelligence agencies consider both theories plausible

Spread:

Despite efforts to contain the virus locally, it spread to other countries by January 2020. The first case outside of China was reported in Thailand, and soon after, cases appeared in Japan and South Korea. The virus reached the United States by January 2020, with the first confirmed case in Washington State

Global Impact :

The World Health Organization (WHO) declared the outbreak a Public Health Emergency of International Concern on January 30, 2020, and a pandemic on March 11, 2020. The pandemic led to unprecedented international lockdowns, travel restrictions, and the implementation of various public health measures to control the spread

Economic and Social Effects:

The pandemic had far-reaching consequences beyond health. Economies around the world were impacted due to lockdowns and reduced consumer activity. The crisis also highlighted and exacerbated existing social inequalities and brought attention to the importance of robust healthcare systems

Vaccines and Treatment:

A significant milestone in combating the pandemic was the development of vaccines. The first COVID-19 vaccines were administered in December 2020, less than a year after the virus was identified. This rapid development was a result of global collaboration and the application of years of research in vaccine technology.

As of my last update in 2021, the pandemic continued to evolve with the emergence of new variants, ongoing vaccination efforts, and discussions about the long-term societal changes brought about by the pandemic. The full impact of COVID-19 will likely be studied for many years to come.

For the most current information, please refer to reliable sources such as the WHO or the CDC, as my knowledge is based on information available up to 2021, and the situation has evolved since then.

- ## COVID-19 Mortality Statistics and Contributing Factors:

Since the outbreak of COVID-19 in late 2019, the world has witnessed significant impacts on public health and mortality rates. By mid-2023, over 6 million deaths have been reported globally due to the virus, with variations in mortality rates observed across different regions and demographics. Here are some key statistics:

1-Global Death Toll: As of mid-2023, the global death toll stands at over 6 million, with significant variations between countries.

2-Regional Variations: Mortality rates have varied significantly, with higher rates observed in countries with older populations and strained healthcare systems**.**

3-Demographic Disparities: Older adults, particularly those aged 65 and above, have been at a significantly higher risk of death. Males have also shown higher mortality rates compared to females

4-Wave Patterns: Different waves of the pandemic have shown varying mortality rates, often influenced by the emergence of new variants, vaccination coverage, and public health measures.

- ## Contributing Factors to COVID-19 Mortality:

1-Age: Older age is the most significant risk factor. The immune system weakens with age, making it harder for older adults to combat infections.

2-Pre-existing Health Conditions: Individuals with underlying health conditions such as cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to experience severe symptoms and death.

3-Access to Healthcare: The quality and accessibility of healthcare services play a crucial role. Countries with better healthcare infrastructure and resources have generally reported lower mortality rates

4-Vaccination Status: Vaccinated individuals have shown significantly lower mortality rates compared to those who are unvaccinated. Vaccines have proven effective in reducing the severity of symptoms and preventing deaths.

5-Socioeconomic Factors: Lower socioeconomic status is associated with higher mortality rates. This is due to factors such as crowded living conditions, lack of access to healthcare, and greater exposure to the virus due to essential work.

6-Genetic Factors: Emerging research suggests that genetic predispositions may influence the severity of COVID-19 and its associated mortality risk.

1.2 The effects of heart disease

The effects of heart disease can be profound and wide-ranging, impacting both physical health and overall quality of life. Here are some of the key effects:

1-Decreased Physical Functioning: Heart disease can lead to symptoms such as chest pain (angina), shortness of breath, fatigue, and weakness, which can significantly reduce a person's ability to perform everyday activities, including work, exercise, and household chores.

2-Reduced Quality of Life: Living with heart disease often requires lifestyle modifications, such as dietary restrictions, medication regimens, and limitations on physical activity. These changes can impact social interactions, leisure activities, and overall enjoyment of life.

3-Increased Risk of Complications: Heart disease increases the risk of various complications, including heart attack, stroke, heart failure,arrhythmias,and sudden cardiac arrest. These complications can have serious consequences, including disability, hospitalization, and death.

4-Emotional Impact: Dealing with a chronic condition like heart disease can be emotionally challenging. It can lead to feelings of anxiety, depression, fear, and stress, both for the individual with the condition and for their loved ones. Adjusting to lifestyle changes, coping with symptoms, and facing uncertainty about the future can all contribute to emotional distress.

5-Financial Burden: Heart disease can impose a significant financial burden on individuals and families due to medical expenses, including doctor visits, hospitalizations, medications, and procedures. It can also lead to lost income and productivity, especially if the individual is unable to work due to their condition or if caregiving responsibilities impact employment.

6-Mortality: Perhaps the most serious effect of heart disease is its potential to be life-threatening. Heart disease remains one of the leading causes of death globally, accounting for millions of deaths each year. While advances in treatment and prevention have improved outcomes for many individuals with heart disease, it still poses a significant risk to health and longevity.

• Types of Heart Disease

Heart disease refers to a range of conditions that affect the heart, including coronary artery disease, heart rhythm problems (arrhythmias), heart defects present at birth (congenital heart defects), and others. It's a leading cause of death globally, accounting for millions of deaths each year.

1. Coronary Artery Disease (CAD): This is the most common type of heart disease and is caused by the buildup of plaque in the coronary arteries, which supply blood to the heart muscle. Plaque is made up of cholesterol, fatty substances, calcium, and other materials found in the blood. Over time, plaque buildup narrows the arteries, reducing blood flow to the heart and increasing the risk of heart attack and stroke.

2. Heart Attack (Myocardial Infarction):A heart attack occurs when the blood flow to a part of the heart is blocked for a long enough time that part of the heart muscle is damaged or dies. It usually happens because of a blood clot that forms in one of the coronary arteries due to plaque buildup.

3. Heart Failure: Heart failure occurs when the heart is unable to pump enough blood to meet the body's needs. It can result from conditions that overwork the heart, such as high blood pressure, coronary artery disease, or diseases that affect the heart muscle itself (cardiomyopathy).

4. Arrhythmias: Arrhythmias are abnormal heart rhythms that can cause the heart to beat too slowly (bradycardia), too quickly (tachycardia), or irregularly. They can occur due to problems with the heart's electrical system, such as damage to the heart muscle from a heart attack or other conditions.

5. Valvular Heart Disease: This refers to conditions affecting the heart valves, which control the flow of blood through the heart. Valvular heart disease can involve valve stenosis (narrowing), valve regurgitation (leaking), or a combination of both. It can be caused by congenital defects, infections, or other factors.

6. Congenital Heart Defects: These are structural abnormalities present in the heart at birth. They can affect the heart's chambers, valves, or blood vessels. Some defects may be minor and not cause any problems, while others can be severe and require surgical intervention.

- ## Risk Factors and Management

Risk factors for heart disease include high blood pressure, high cholesterol, smoking, obesity, diabetes, a sedentary lifestyle, unhealthy diet, excessive alcohol consumption, and a family history of heart disease. Prevention and management strategies include lifestyle modifications (e.g., regular exercise, healthy diet, smoking cessation), medication (e.g., statins, blood pressure medications), and in some cases, surgical interventions (e.g., angioplasty, bypass surgery, valve replacement). Early detection and treatment are crucial for improving outcomes and reducing the risk of complications associated with heart disease. Regular medical check-ups, screening tests, and adherence to treatment plans are essential for individuals at risk or already diagnosed with heart disease.

1.3 What is an ECG (Electrocardiogram)?

An Electrocardiogram (ECG or EKG) is a non-invasive diagnostic tool used to assess the electrical activity of the heart. It is a fundamental and widely used procedure in cardiology, providing valuable information about the heart's rhythm, electrical conduction, and overall cardiac health.

- ## How ECG Works

The heart generates electrical impulses that regulate the contraction of its muscles, ensuring effective pumping of blood throughout the body. An ECG measures these electrical signals via electrodes placed on the skin. Typically, a standard ECG involves 12 leads, providing different perspectives of the heart's electrical activity from various angles.

Components of an ECG:

An ECG trace consists of several key components:

P Wave: Represents atrial depolarization, the electrical activity associated with the contraction of the atria.

QRS Complex: Corresponds to ventricular depolarization, the electrical activity related to the contraction of the ventricles. It is the most prominent part of the ECG trace.

T Wave: Represents ventricular repolarization, the process of the ventricles recovering their electrical state before the next contraction.

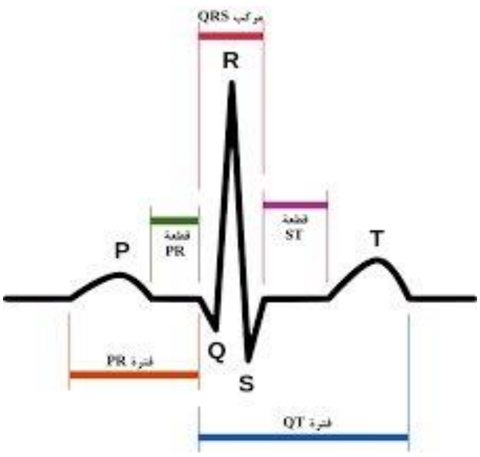


figure (1)

- **Clinical Significance**

1-Heart Rate and Rhythm: An ECG is crucial for determining the heart rate and identifying arrhythmias, such as atrial fibrillation, bradycardia, tachycardia, and other irregular heartbeats.

2-Conduction Abnormalities: It helps in diagnosing issues with the electrical conduction pathways of the heart, such as bundle branch blocks and heart blocks.

3-Myocardial Ischemia and Infarction: An ECG can detect signs of reduced blood flow to the heart muscle (ischemia) and identify patterns indicative of a heart attack (myocardial infarction), such as ST-segment elevation or depression.

4-Electrolyte Imbalances: Changes in the ECG can reflect imbalances in electrolytes, such as potassium and calcium levels, which are crucial for normal cardiac function.

5-Structural Abnormalities: It can provide indirect evidence of structural issues like left ventricular hypertrophy or other cardiomyopathies.

An ECG is a vital tool in the diagnosis and management of heart diseases, offering a non-invasive, quick, and informative method to evaluate the electrical activity of the heart. Its ability to provide real-time insights into cardiac function makes it indispensable in both clinical and emergency settings

● Advantages of ECG:

1. **Non-invasive and Painless**:

   ECG is a non-invasive test, meaning it does not require any needles or incisions. It is painless, making it suitable for patients of all ages.

2. **Quick and Easy to Perform**:

   An ECG can be completed in a few minutes, allowing for rapid assessment of a patient's heart condition.

3. **Cost-effective**:

   Compared to more complex imaging techniques, ECG is relatively inexpensive, making it accessible in a wide range of healthcare settings.

4. **Immediate Results**:

   ECG provides real-time data, allowing healthcare professionals to quickly detect and respond to cardiac events or abnormalities.

5. **Portable**:

   Portable ECG devices are available, enabling monitoring and diagnosis in remote locations or during emergencies.

6. **Valuable Diagnostic Tool**:

   ECG can diagnose a variety of heart conditions such as arrhythmias, myocardial infarction (heart attack), and other heart-related issues.

7. **Baseline Data**:

   Regular ECGs can provide baseline data for a patient, helping to track changes over time and manage chronic conditions effectively.

- ## Disadvantages of ECG:

1. **Limited Diagnostic Scope**:

   While ECG is effective for diagnosing many heart conditions, it cannot detect all types of heart diseases or issues. Some abnormalities might not show up on an ECG.

2. **Requires Skilled Interpretation**:

   Accurate interpretation of ECG results requires skilled healthcare professionals. Misinterpretation can lead to incorrect diagnoses or treatment.

3. **Potential for False Positives/Negatives**:

   ECG can sometimes produce false positives (indicating a problem where there is none) or false negatives (failing to detect an existing problem), leading to unnecessary anxiety or missed diagnoses.

4. **Not Suitable for All Patients**:

   Certain patients, such as those with obesity or significant chest hair, might have difficulty obtaining clear readings, which can affect the accuracy of the results.

5. **Only Reflects Current Heart Activity**:

   ECG provides a snapshot of the heart's activity at the moment of the test. It may miss intermittent or transient issues that do not occur during the test.

6. **Limited Structural Information**:

   ECG focuses on the electrical activity of the heart and provides limited information about the heart's structural aspects. Other imaging techniques like echocardiography or MRI might be needed for a more comprehensive assessment.

1.4 motivations:

Motivation Behind Developing an ECG-Based Diagnostic Application

The motivation behind developing an application utilizing ECG images for diagnosing heart diseases and COVID-19 stems from the pressing need to enhance early diagnosis and provide effective healthcare interventions.

1.Early Diagnosis and Effective Healthcare:

Identifying Precise Changes:

This technology offers the capability to identify precise changes in heart activity, enabling the early detection of subtle abnormalities indicative of heart diseases.

Capturing Early Signs:

By capturing early signs of heart diseases, such as arrhythmias or myocardial infarction, our application facilitates timely medical intervention, potentially preventing adverse outcomes and complications.

Understanding Cardiac Effects of COVID-19:

Furthermore, the incorporation of ECG analysis in diagnosing COVID-19 allows for a comprehensive assessment of its cardiac effects. Early detection of cardiac involvement in COVID-19 patients can guide appropriate treatment strategies and improve patient outcomes.

2-Quicker Treatment Decisions and Enhanced Preventive Measures:

Quicker Treatment Decisions:

Rapid identification of cardiac abnormalities through ECG analysis enables healthcare providers to make quicker treatment decisions, minimizing delays in initiating appropriate therapies and improving patient prognosis.

Improving Recovery Chances:

Timely diagnosis and intervention contribute to improved recovery chances, as early treatment is often associated with better outcomes and reduced morbidity.

Enhancing Preventive Measures:

Moreover, early detection of cardiac abnormalities allows for the implementation of preventive measures, such as lifestyle modifications or medication management, to mitigate the progression of heart diseases and reduce the risk of future cardiac events.

Goal of Developing a Simple and Effective Diagnostic Tool:

Our ultimate goal as researchers is to develop a simple yet highly effective diagnostic tool utilizing artificial intelligence technology. By leveraging the power of AI, we aim to streamline the diagnostic process, assist healthcare providers in making faster and more accurate diagnoses, and ultimately improve patient care outcomes**.**

1.5 Problem Statement

- **The Importance of Developing a Website for Accurate and Efficient Heart Disease Diagnosis**
  When diagnosing whether an individual is infected with the coronavirus (COVID-19) or has a heart condition, medical professionals face significant challenges due to the limitations of the two primary diagnostic protocols: PCR testing and lung imaging. These constraints impact the effectiveness and practicality of these methods in clinical settings. Understanding these challenges highlights the necessity of developing a more accurate and efficient diagnostic tool, such as a dedicated website for diagnosing heart diseases.

1. **Challenges in PCR Testing:**

Overview: Polymerase Chain Reaction (PCR) testing is a molecular diagnostic technique used to detect the genetic material of the coronavirus. It is widely regarded as the gold standard for COVID-19 diagnosis due to its high sensitivity

Limitations:

False Negatives:

Accuracy Concerns: Despite its high sensitivity, PCR tests can sometimes yield false-negative results, particularly if the viral load in the sample is low or if the sample collection technique is suboptimal. This can lead to misdiagnosis and delayed treatment, posing a risk of further transmission.

Turnaround Time:

Processing Delays: Although PCR tests are designed to provide quick results, the actual turnaround time can vary. In many cases, it takes several hours to process the samples in a laboratory setting. Delays in obtaining results can impede timely decision-making and isolation measures.

Resource Intensive:

Infrastructure Requirements: PCR testing requires specialized equipment, reagents, and trained personnel. This makes it resource-intensive and potentially inaccessible in resource-limited settings. Additionally, scaling up PCR testing to meet high demand can strain healthcare resources.

2.Challenges in PCR Lung Imaging Testing

Overview: Lung imaging, including chest X-rays and CT scans, is used to visualize the lungs and detect abnormalities. It is often employed to assess the extent of lung involvement in COVID-19 and to identify heart conditions that may present with similar symptoms.

Limitations:

1-Cost:

Expensive Procedure: Lung imaging is a costly diagnostic tool. The expenses associated with imaging can be a barrier, particularly for uninsured patients or in healthcare systems with limited funding. High costs may limit widespread access to this diagnostic method.

2-Time-Consuming:

Extended Duration: The process of conducting lung imaging and interpreting the results is time-consuming. This includes the time required for the patient to undergo the procedure, as well as the time needed for a radiologist or physician to analyze and report the findings. This can delay diagnosis and treatment initiation.

3-Need for Specialist Intervention:

Requirement for Medical Expertise: Lung imaging necessitates the involvement of specialized healthcare professionals. Radiologists and physicians must interpret the imaging results, which adds an additional layer of complexity and dependency on healthcare personnel. In emergency situations or in regions with a shortage of specialists, this requirement can hinder prompt diagnosis and care.

1.6 solution statement:

- **Innovative Solution: ECG-Based Diagnosis Website**

In response to the limitations associated with PCR testing and lung imaging in diagnosing diseases like COVID-19 and heart conditions, we have developed an innovative solution: a website that utilizes Electrocardiogram (ECG) images for diagnosis.

How It Works

ECG Analysis:

- Our website employs advanced algorithms to analyze ECG images uploaded by users.
- ECGs provide valuable insights into cardiac activity, helping detect heart conditions such as arrhythmias, myocardial infarction, and heart failure.

Quick and Accurate Results:

- By utilizing ECG images, our method offers rapid and precise diagnosis without the need for time-consuming PCR testing or costly lung imaging procedures.
- Users receive instant feedback, saving them valuable time and effort.

Cost-Effective Solution:

- Unlike lung imaging, which can be expensive, our website provides an affordable alternative for disease diagnosis.
- Users can access the service without the financial burden
- associated with traditional diagnostic methods.

Guided Treatment and Prevention:

- With swift diagnosis facilitated by our website, users can initiate treatment promptly, improving their chances of recovery and preventing complications.
- Early detection of heart conditions allows for timely intervention and implementation of preventive measures, enhancing overall cardiac health

- **Benefits:**

1. **Time-Saving:**

   Our website streamlines the diagnostic process, allowing users to receive prompt results without delays associated with PCR testing or imaging procedures.

2. **Cost-Effective:**

   By eliminating the need for expensive imaging techniques, our solution offers a budget-friendly option for disease diagnosis.

3. **Accurate Diagnosis:**

   Leveraging ECG images ensures accurate and reliable diagnosis, enhancing confidence in treatment decisions.

4. **Improved Outcomes:**

   Early diagnosis facilitated by our website leads to faster treatment initiation, ultimately improving patient outcomes and reducing the risk of complications.

Our ECG-based diagnosis website addresses the limitations of traditional diagnostic methods like PCR testing and lung imaging. By harnessing the power of technology, we provide users with a convenient, cost-effective, and accurate solution for diagnosing diseases. This innovative approach not only saves time, effort, and money but also improves treatment outcomes and promotes preventive healthcare practices.

1.7 scope and target audience:

- **Scope:**

The scope of this website encompasses individuals seeking rapid and accurate diagnosis of heart diseases and COVID-19 using Electrocardiogram (ECG) images. It includes the development and implementation of advanced algorithms for ECG analysis, as well as the provision of user-friendly interfaces for uploading and interpreting ECG data. Additionally, the website will facilitate seamless communication between users and healthcare professionals for further consultation and treatment planning.

- **Target Audience:**
  1. **Patients and Caregivers:**

     Individuals experiencing symptoms of heart diseases or COVID-19, as well as their caregivers seeking timely diagnosis and appropriate medical intervention.

  2. **Healthcare Professionals:**

     Cardiologists, general practitioners, and other healthcare providers interested in leveraging technology for efficient diagnosis and management of cardiovascular conditions and COVID-19 cases.

  3. **Medical Researchers:**

     Researchers and academics involved in cardiovascular health and infectious disease studies, who may utilize the website for data collection, analysis, and collaborative research initiatives.

  4. **Healthcare Institutions:**

     Hospitals, clinics, and medical centers are looking to integrate innovative diagnostic tools into their healthcare delivery systems to enhance patient care and outcomes.

  5. **Public Health Authorities:**

     Government agencies and organizations responsible for public health surveillance and management of infectious diseases, seeking efficient and scalable solutions for disease detection and monitoring.

  6. **Technology Developers:** Innovators and developers in the field of healthcare technology, interested in contributing to the advancement of diagnostic algorithms and digital health solutions**.**

Chapter 02

Related Work

In the past few decades, the research community has focused on artificial intelligence by working in digital image processing, computer vision, and machine learning to provide a platform between human and machine theory .

His work is widely recognized by several companies and medical fields for classification, detection, and identification of cardiac disorder, which play a vital role in the health community. Techniques used for the identification of cardiac disorder using deep learning have been focused on for many years.

In recent studies, the state-of-the-art research on deep learning gained the potential growth with satisfactory results in detection and classification tasks on medical images

Most studies treated dimensional ECG signals as time-series classification by keeping in the view of deep learning.

For example, PCGs and ECGs are used to diagnose heart disease.

2.1 Literature Review on ECG Signal :

Literature Review on ECG Signals

Authors & Year	Dataset & classes & Pre-processing & Feature Extraction	Classification	Results
Mazaher and khodadadi 2020	MIT-BIH Arrhythmia 7 classes 71 F Gain and fixed element reduction + rescaling and standardization + Morphological + Frequency + Non-linear features	KNN, FF net, Fit net, RBFNN Pat net	10 Fold cross-validation A = 98.75% using Pat net
Khalil and Adib 2020	MIT-BIH Arrhythmia 6 classes 45000 F Normalization ML-WCNN (1D-CNN + SWT)	Softmax	10 Fold cross validation A = 99.75%
Shaker et al. 2020	MIT-BIH Arrhythmia 15 classes 102098 HB Butterworth band pass filter + GAN Deep learning approach based on the Inception Module	Softmax	Train: 21077 HB Test: 81021 HB P = 90. 0% A = 98.0% SPEC = 97.4% SEN = 97.7%
Tyagi and Mehra 2021	MIT-BIH Arrhythmia Database 16 classes 110100 HB DWT CNN-GOA	Softmax	Training = 73006 HB Test = 37094 HB A = 99.58% Error Rate = 0.42%
Eltrass et al. 2021	MIH-BIH ARR dataset + MIT-BIH NSR dataset + BIDMC CHF dataset 3 classes 972 F Normalization + multi-stage kernel adaptive filter design CNN (Alexnet) + CQ-NSGT + MLP	Softmax	k- fold cross validation A = 98.82% SPEC = 99.21% SEN = 98.87% P = 99.20%

Literature Review on Changes in ECG for COVID

Author / Year	Number of COVID Patients	Main Changes in ECG due to COVID disease
Wang et al. 2020	201 out of 319	Abnormal ECG Changes in ST-T Various Heart Diseases: AF, Atrial Tachycardia, Abnormal Q-Wave, Weak R-wave progression, RBBB
Pavri et al. 2020	75	50.7% of patients have shortened PR interval and acceleration in the HR. 49.3% of the patients have no change in their ECG.
Angeli et al. 2020	50	30% have ST-T abnormality 30% have left ventricular hypertrophy Others have AF, tachy-brady syndrome, acute pericarditis, and rare RBBB and MI
Li et al. 2020	113 50 died 63 survived	Ventricular arrhythmia and Sinus tachycardia existed in the people who died and survived
Santoro et al. 2020	110	Prolongation in the QT segment of ECG in the COVID patients
Jain et al. 2020	2006	Prolongation in the QT segment of ECG in the COVID patients
McCullough et al. 2022	756	Atrial premature contractions, Intraventricular block, Repolarization abnormalities, RBBB and rare ST-elevation
Lam et al. 2020	18	PR prolongation and depression, Atrial flutter, RBBB, biphasic T-wave Problems in the Q-wave, 63% of the patients suffer from atrial trigeminal
Bertini et al. 2020	431	22% suffer from AF 30% suffer from Acute right ventricular ST-T prolongation in 4 patients

Literature Review on COVID and ECG arrhythmias Diagnosis based on ECG Image reports

Authors	Methodology	Experiments	Results
Anwar et al. 2021	EfficientB0 + Softmax	**Multi-Class Experiment** COVID (250) vs MI (77) vs PMI (203) vs Abnormal (548) vs Normal (859)	Five-Fold Cross-validation A = 81.8 without augmentation A = 77.6 with augmentation
Shahin et al. 2021	Vgg16 + Softmax	**Multi-Class Experiment** **5- class diagnosis** COVID (250) vs MI (77) vs PMI (203) vs Abnormal (548) vs Normal (859)	K-Fold Cross-validation A = 81.39
Rahman et al. 2022	DenseNet201 + Softmax	**Multi-Class Experiments** 1st: COVID (250) vs MI (77) vs PMI (203) vs Abnormal (548) vs Normal (859) 2nd: COVID (250) vs Normal (250) vs Abnormalities (250) **Binary-Class Experiment** 3rd: COVID (250) vs Normal (250)	10-Fold cross-validation 1st: A= 97.83 2nd: A = 97.36 3rd: A= 99.10
Attallah 2021	Fully connected features of 5 pre-trained models + DWT of the max-pooling features + 3 classifiers	**Multi-Class Experiment** 1st: COVID (250) vs Normal (250) vs Abnormalities (250) **Binary-Class Experiment** 2nd: COVID (250) vs Normal (250)	10-Fold Cross-validation 1st: A = 91.73% 2st: A = 98.80% Using voting classifier
Attallah 2022	Fully connected features of 10 pre-trained models + feature selection + classification using six classifiers	**Multi-Class Experiment** 1st: COVID (250) vs Normal (250) vs Abnormalities (250) **Binary-Class Experiment** 2nd: COVID (250) vs Normal (250)	10-Fold Cross-validation 1st: A = 91.60% using RF classifier 2st: A = 98.20% using DT classifier

Literature Review on 2D ECG Image Reports

Authors	No classes and leads	Types of ECG classes	Number of ECG image Reports or Recorded Images	Methodology	Results
Mohamed et al. 2015	2 classes 12 ECG leads	Normal vs Abnormal	45 ECG Image leads	Binarization + Hear features + ANN	A = 99%
Hao et al. 2019	2 classes 12 ECG leads	Normal vs MI	1557 ECG image reports	Text detection of lead label + shallow neural network + depth fusion	Train and validation: 957 Test: 600 A= 94.73 % Sen = 96.41% Spec = 95.94% F1-Score = 93.79%
Ferreira et al. 2019	2 classes 12 ECG leads	Normal vs Abnormal	100673 image samples	LBP + MLP	A = 96.18%
Du et al. 2020	20 classes 12- ECG leads	20 ECG class types	1st dataset: 27,820 image reports. 2nd dataset: 33000 image reports	Discovery of weak supervised and spatial attention part + recurrent label inference	1st dataset: A = 79.23% F1-Score: 73.88% 2nd dataset: A= 90.42% F1-Score: 86.87%
Ribeiro et al. 2020	6 classes 12 – ECG leads	1dAVb, RBBB,LB BB,SB, AF, and ST	2,322,513 ECG Image leads	DNN	98% Training and validation = 2% F1-Score over 80% on the validation data

2.2 Continued Literature Review on ECG Signal:

Continued Literature Review on COVID and ECG arrhythmias Diagnosis based on ECG Image reports

Authors	Methodology	Experiments	Results
Ozdemir et al. 2021	GLCM + Hexeixal Mapping + a proposed deep learning model + Softmax	**Binary-Class Experiments** 1st: COVID (250) vs Normal (250) 2nd: Positive (COVID (250)) vs Negative (Norm al (250))	5-Fold Cross-validation 1st: A = 96.20% 2st: A = 93.20%
Sobahi et al. 2022	A proposed 17 DL layers model based on a 3D convolution netw ork + Softmax	**Multi-Class Experiments** 1st: COVID (250) vs Normal (250) vs Abnormal (250) **Binary-Class Experiment** 2nd: COVID (250) vs Normal (250)	10-Fold cross-validation 1st: A= 92.0 2nd: A = 99.0
Irmak 2022	A proposed 18 DL model layers + Softmax	**Multi-Class Experiments** 1st: COVID (218) vs MI (77) vs Abnormal (500) vs Normal (713) 2st: COVID (218) vs MI (77) vs Abnormal (500) **Binary-Class Experiments** 3rd: COVID (218) vs MI (77) 4th: COVID (218) vs Abnormal (500) 5th: COVID (218) vs Normal (713)	1st: A= 82.00 2nd: A = 86.60 3rd: A= 96.70 4th: A = 93.20 5th: A= 98.60

Continued Literature Review on ECG Signals

Authors & Year	Dataset & classes & Pre-processing & Feature Extraction	Classification	Results
Panganiban, et al. 2021	MIT-BIH AF, PAF Prediction Challenge, PTB, Challenge 2015Training set, and Fantasia databases 5 classes 323 Records Records are cut and converted to text files in the form of samples in CSV files CWT + InceptionV3	Softmax	Binary class: Total : 4248 Spectrogram Images A = 98.73% PPV: 96.83% Multi-class: Total : 4248 Spectrogram Images A= 97.33% F1-Score = 99.21%
Dixit and Kala 2022	CHF-RR, BIDMC-CHF, Fantasia, MIT-BIH-NSR datasets 2 classes 25498 F Normalization + DCNN-LSTM- AM	LSTM	A = 99.52% SEN = 99.31% SPEC = 99.28% F-Score = 98.94% AUC = 99.9%
Anand et al. 2022	PTB-XL dataset 5 classes 27826 R + Arrhythmia dataset 4 classes 10588 R Normalization + ST-CNN-GAP-5	Sigmoid	PTB-XL: A = 89.73% AUC = 93.41% Arrhythmia: A = 95.84% AUC = 99.46%
Shin et al. 2022	MIT-BIH database 4 classes 654 R Resizing + digital filter + Data Augmentation + Scalogram MobileNetV2 + BiLSTM	Softmax	A = 91.7% SEN = 92.0% SPEC = 91.0% Precision = 92.0% F1-Score = 92.0%

Continued Literature Review on 2D ECG Image Reports

Authors	No classes and leads	Types of ECG classes	Number of ECG image Reports or Recorded Images	Methodology	Results
Gliner et al. 2020	8-classes 12- ECG leads	AF, I-AVB, LBBB, RBBB, PAC, PVC, STD, and STE	41830 Digital and Image leads	CNN-ima and CNN-dig	Training and validation = 86% Testing = 14% A = 88%
Xie et al. 2021	2 classes 12-ECG leads	Normal, Stroke	98 ECG image reports	Proposed Dense Net architecture	Train: 90 Test: 8 A = 85.82%
Khan et al. 2021	4 classes 12- ECG leads	Normal, Abnormal, MI, and PMI	11,148 ECG Image Leads	Data resizing and labeling + SSD + Mo bile v2	80% Training 20% Test A = 98.33%
Bassiouni et al. 2022	4 classes 12 - ECG leads	Normal, Abnormal, MI, and PMI	922 ECG image reports	Xception with TCN	Five Fold Cross-validation A = 98.6%
Bridge, J. et al. 2022	2 classes 12 – ECG leads	Sinus rhythm, abnormal rhythm	1172 ECG image reports	InceptionV3	Train and Validation: 1000 Test: 172 Sen: 80.16% Spec: 99.3%

Chapter03

Dataset

3.1 Data resources

The first Dataset:

The dataset is composed of ECG images of cardiac and COVID patients.

There are total 1937 images collected using 12 lead ECG devices from different healthcare and cardiology institutes of Pakistan,

The prominent ones are Ch. Pervaiz Elahi Institute of Cardiology Multan, Nishtar Medical University Multan and Punjab Institute of Cardiology Lahore

The second Dataset:

steps involved in collecting the 12-lead-based ECG Images data required to develop the system Data resource identification was the main and primary challenge for the cardiac disease detection system.

Consequently, in the first phase, various sites, repositories were explored.
However, in spite of all efforts, all publicly available datasets that can be collected are time series data.

Nevertheless, those available datasets are inconsistent with the data needed for this study.

(e manual collection of 12-lead-based standard ECG images directly from the cardiac institute is a difficult task.

(e authors approached one of the best cardiac institutes, i.e., Ch. Pervaiz Elahi Institute of Cardiology Multan, Pakistan, for data
collection.

(e manual collection of ECG images from the cardiac institute took several months.

(e ECG images collected have been endorsed and annotated by numerous cardiologists and reexamined by the domain experts who have experience in ECG interpretation.

3.2 Data description

The first Dataset:

The dataset has 5 categories as shown in table

Category	No of patients
Covid_19	**250**
Myocardial Infarction	**77**
History Myocardial Infarction	**203**
Abnormal Heartbeat	**548**
Normal	**859**

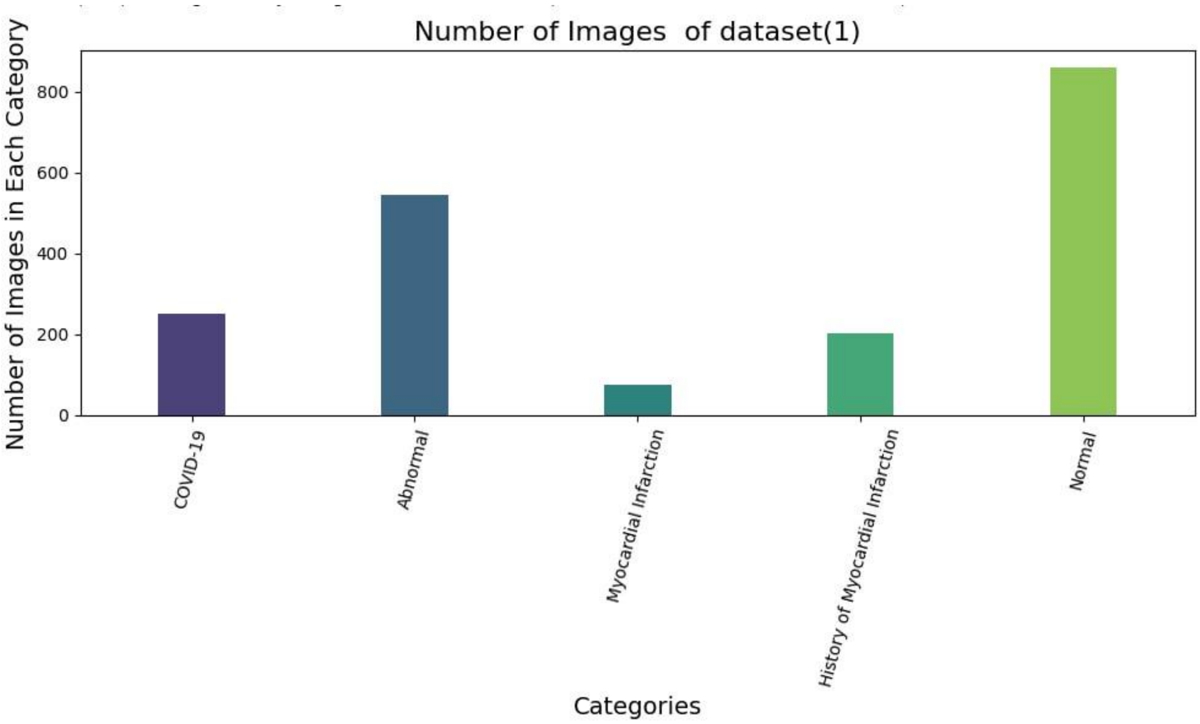


figure (2)

1) COVID:

COVID-19 is severe respiratory syndrome caused by SARS virus.

This virus has killed more than 2 million people in one year after its outbreak worldwide.

They are transmitted through cough and breath by the patients to normal individuals.

Polymerase chain reaction test (PCR) is commonly used for its detection as a standard way.

2) Myocardial Infarction:

Myocardial Infarction (MI) is a severe cardiac disease occurring due to blockage of blood supply to parts of the heart.

It results in damage to the heart muscle and causes heart attack. In acute cases it leads to death of the individual

3) History of Myocardial Infarction:

The images of the patients who have been recovered from MI or has previous history of MI are placed in the category.

4) Abnormal Heartbeat:

These patients have shortness of breath and respiratory problem after recovering from MI and COVID.

5) Normal:

These individuals don’t have any cardiac related disease or any kind of other abnormality and health issue.

The second Dataset:

Table shows the total numbers of Images used for cardiac disorder detection used for each class.

Class	12-lead ECG	Total images
Myocardial infarction (MI)	**240**	**2880**
Abnormal heartbeat	**233**	**2796**
Previous history of MI	**172**	**2064**
Normal	**284**	**3408**

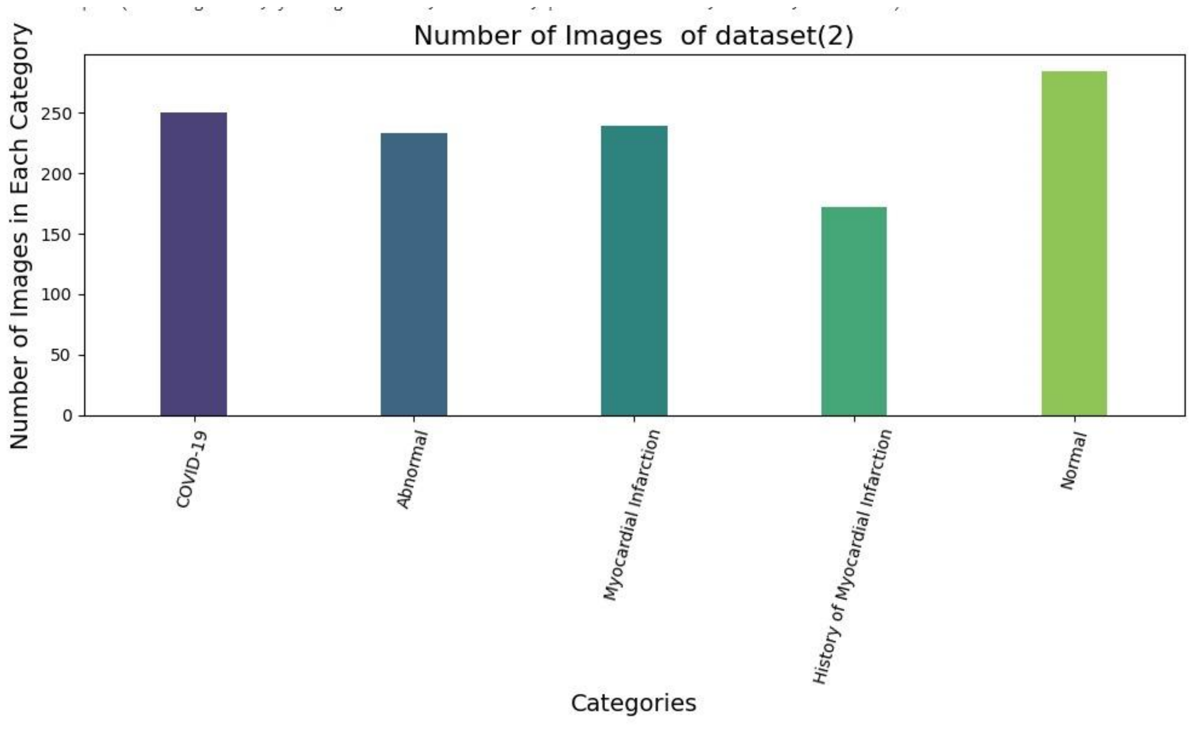
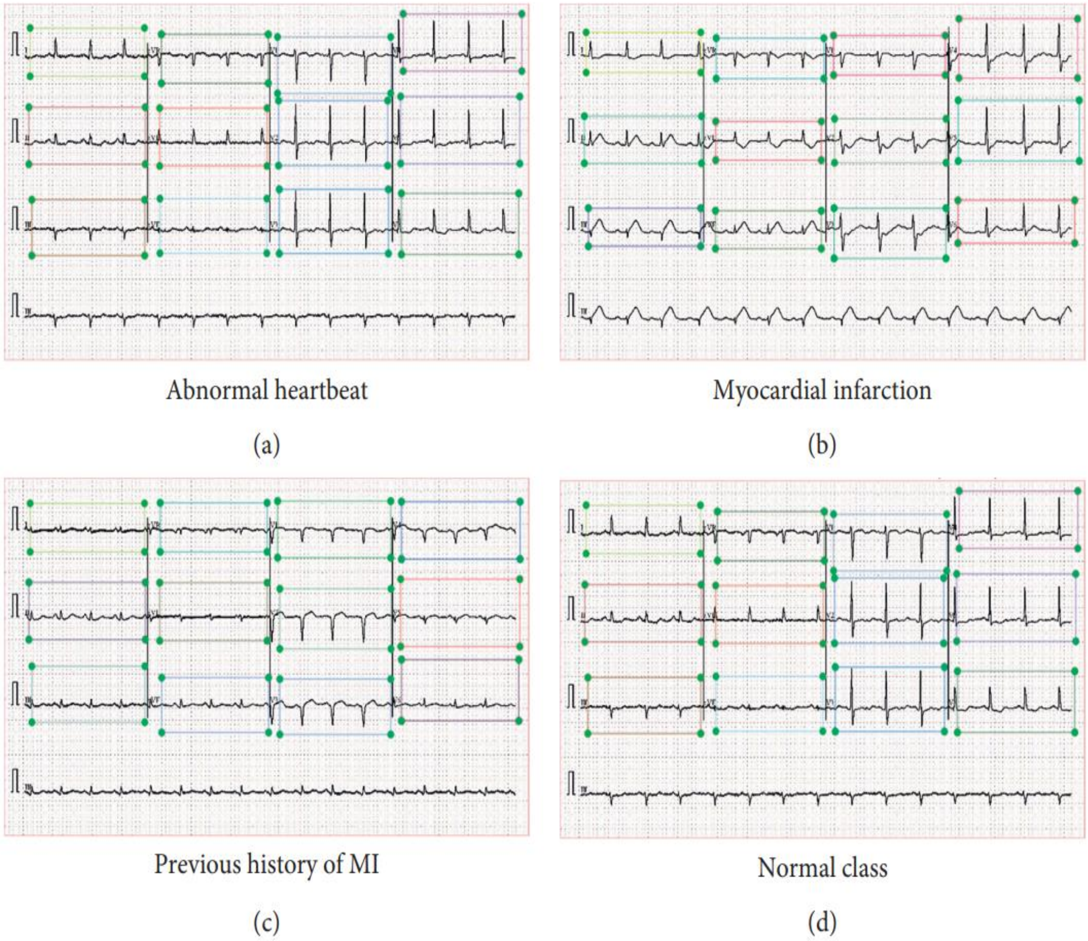


figure (3)



Abnormal heartbeat

(a)

Myocardial infarction

(b)

Previous history of MI

(c)

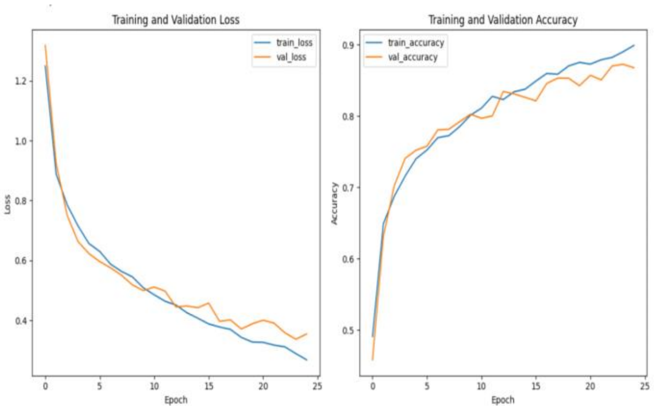
Normal class

(d)

figure (4)

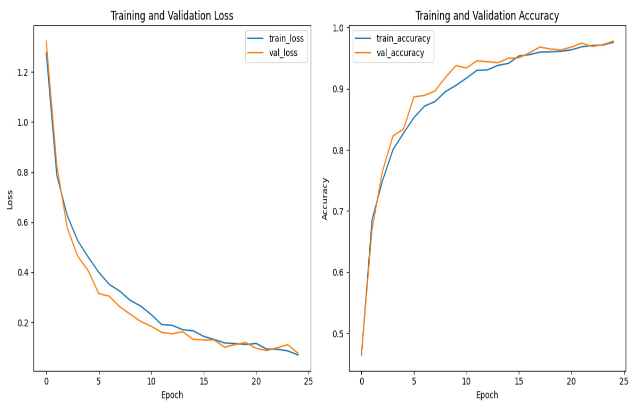
3.3 the comparison between two datasets:

the first dataset




```
179/179 [==============================] - 207s 1s/step - loss: 0.4856 - accuracy: 0.8112 - val_loss: 0.5117 - val_accuracy: 0.7971 - lr: 1.0000e-05
Epoch 12/25
179/179 [==============================] - 205s 1s/step - loss: 0.4642 - accuracy: 0.8278 - val_loss: 0.4979 - val_accuracy: 0.8003 - lr: 1.0000e-05
Epoch 13/25
179/179 [==============================] - 204s 1s/step - loss: 0.4514 - accuracy: 0.8233 - val_loss: 0.4451 - val_accuracy: 0.8346 - lr: 1.0000e-05
Epoch 14/25
179/179 [==============================] - 203s 1s/step - loss: 0.4258 - accuracy: 0.8343 - val_loss: 0.4484 - val_accuracy: 0.8313 - lr: 1.0000e-05
Epoch 15/25
179/179 [==============================] - 204s 1s/step - loss: 0.4075 - accuracy: 0.8378 - val_loss: 0.4424 - val_accuracy: 0.8264 - lr: 1.0000e-05
Epoch 16/25
179/179 [==============================] - 204s 1s/step - loss: 0.3884 - accuracy: 0.8493 - val_loss: 0.4579 - val_accuracy: 0.8215 - lr: 1.0000e-05
Epoch 17/25
179/179 [==============================] - 202s 1s/step - loss: 0.3782 - accuracy: 0.8599 - val_loss: 0.3971 - val_accuracy: 0.8460 - lr: 1.0000e-05
Epoch 18/25
179/179 [==============================] - 216s 1s/step - loss: 0.3701 - accuracy: 0.8589 - val_loss: 0.4022 - val_accuracy: 0.8533 - lr: 1.0000e-05
Epoch 19/25
179/179 [==============================] - 216s 1s/step - loss: 0.3435 - accuracy: 0.8704 - val_loss: 0.3714 - val_accuracy: 0.8533 - lr: 1.0000e-05
Epoch 20/25
179/179 [==============================] - 204s 1s/step - loss: 0.3281 - accuracy: 0.8755 - val_loss: 0.3888 - val_accuracy: 0.8427 - lr: 1.0000e-05
Epoch 21/25
179/179 [==============================] - 205s 1s/step - loss: 0.3270 - accuracy: 0.8730 - val_loss: 0.4009 - val_accuracy: 0.8574 - lr: 1.0000e-05
Epoch 22/25
179/179 [==============================] - 203s 1s/step - loss: 0.3178 - accuracy: 0.8791 - val_loss: 0.3910 - val_accuracy: 0.8509 - lr: 1.0000e-05
Epoch 23/25
179/179 [==============================] - 204s 1s/step - loss: 0.3119 - accuracy: 0.8825 - val_loss: 0.3587 - val_accuracy: 0.8704 - lr: 1.0000e-05
Epoch 24/25
179/179 [==============================] - 206s 1s/step - loss: 0.2894 - accuracy: 0.8903 - val_loss: 0.3374 - val_accuracy: 0.8729 - lr: 1.0000e-05
Epoch 25/25
179/179 [==============================] - 203s 1s/step - loss: 0.2685 - accuracy: 0.8992 - val_loss: 0.3550 - val_accuracy: 0.8680 - lr: 1.0000e-05
39/39 [==============================] - 32s 807ms/step - loss: 0.3438 - accuracy: 0.8704
Test Loss: 0.3438364565372467
Test Accuracy: 0.8704156279563904
```

the second dataset




```
Epoch 9/25
183/183 [==============================] - 225s 1s/step - loss: 0.2878 - accuracy: 0.8951 - val_loss: 0.2326 - val_accuracy: 0.9185 - lr: 1.0000e-05
Epoch 10/25
183/183 [==============================] - 225s 1s/step - loss: 0.2652 - accuracy: 0.9054 - val_loss: 0.2037 - val_accuracy: 0.9377 - lr: 1.0000e-05
Epoch 11/25
183/183 [==============================] - 222s 1s/step - loss: 0.2311 - accuracy: 0.9172 - val_loss: 0.1844 - val_accuracy: 0.9337 - lr: 1.0000e-05
Epoch 12/25
183/183 [==============================] - 228s 1s/step - loss: 0.1913 - accuracy: 0.9300 - val_loss: 0.1598 - val_accuracy: 0.9457 - lr: 1.0000e-05
Epoch 13/25
183/183 [==============================] - 221s 1s/step - loss: 0.1881 - accuracy: 0.9307 - val_loss: 0.1542 - val_accuracy: 0.9441 - lr: 1.0000e-05
Epoch 14/25
183/183 [==============================] - 219s 1s/step - loss: 0.1707 - accuracy: 0.9381 - val_loss: 0.1627 - val_accuracy: 0.9425 - lr: 1.0000e-05
Epoch 15/25
183/183 [==============================] - 221s 1s/step - loss: 0.1662 - accuracy: 0.9413 - val_loss: 0.1323 - val_accuracy: 0.9497 - lr: 1.0000e-05
Epoch 16/25
183/183 [==============================] - 229s 1s/step - loss: 0.1441 - accuracy: 0.9535 - val_loss: 0.1298 - val_accuracy: 0.9505 - lr: 1.0000e-05
Epoch 17/25
183/183 [==============================] - 222s 1s/step - loss: 0.1317 - accuracy: 0.9555 - val_loss: 0.1296 - val_accuracy: 0.9585 - lr: 1.0000e-05
Epoch 18/25
183/183 [==============================] - 222s 1s/step - loss: 0.1176 - accuracy: 0.9598 - val_loss: 0.1007 - val_accuracy: 0.9681 - lr: 1.0000e-05
Epoch 19/25
183/183 [==============================] - 222s 1s/step - loss: 0.1152 - accuracy: 0.9603 - val_loss: 0.1111 - val_accuracy: 0.9649 - lr: 1.0000e-05
Epoch 20/25
183/183 [==============================] - 218s 1s/step - loss: 0.1116 - accuracy: 0.9610 - val_loss: 0.1198 - val_accuracy: 0.9633 - lr: 1.0000e-05
Epoch 21/25
183/183 [==============================] - 220s 1s/step - loss: 0.1157 - accuracy: 0.9636 - val_loss: 0.0961 - val_accuracy: 0.9681 - lr: 1.0000e-05
Epoch 22/25
183/183 [==============================] - 222s 1s/step - loss: 0.0932 - accuracy: 0.9685 - val_loss: 0.0876 - val_accuracy: 0.9744 - lr: 1.0000e-05
Epoch 23/25
183/183 [==============================] - 220s 1s/step - loss: 0.0925 - accuracy: 0.9706 - val_loss: 0.0985 - val_accuracy: 0.9688 - lr: 1.0000e-05
Epoch 24/25
183/183 [==============================] - 218s 1s/step - loss: 0.0862 - accuracy: 0.9713 - val_loss: 0.1109 - val_accuracy: 0.9720 - lr: 1.0000e-05
Epoch 25/25
183/183 [==============================] - 228s 1s/step - loss: 0.0697 - accuracy: 0.9759 - val_loss: 0.0762 - val_accuracy: 0.9776 - lr: 1.0000e-05
40/40 [==============================] - 229s 6s/step - loss: 0.0922 - accuracy: 0.9721
Test Loss: 0.09224992245435715
Test Accuracy: 0.9720670580863953
```

Chapter04

preprocessing

4.1 What is noise in dataset:

- In the context of datasets and data analysis, "noise" refers to irrelevant or random variations in the data that do not reflect the underlying patterns or relationships of interest. Noise can arise from various sources, including measurement errors, data collection artifacts, or simply inherent randomness in the phenomenon being studied.

- Noise can obscure meaningful patterns in the data, making it more difficult to draw accurate conclusions or build reliable models. Therefore, one of the primary goals of data preprocessing and analysis is often to identify and minimize the impact of noise on the analysis results.

- In the context of images, "noise" refers to random variations in pixel values that are not part of the actual image content. These variations can be introduced during image acquisition, transmission, or processing, and they can degrade the quality of the image by adding unwanted artifacts or reducing the clarity of the visual information.

There are several types of noise that can affect images:

1.**Gaussian Noise:** Gaussian noise follows a Gaussian (normal) distribution and is characterized by its randomness and smoothness. It appears as a subtle graininess or fuzziness in the image.

2.Salt and Pepper Noise: Salt and pepper noise, also known as impulse noise, manifests as randomly occurring bright and dark pixels scattered throughout the image. It can result from errors in image acquisition or transmission.

3.Speckle Noise: Speckle noise is a type of noise that appears as granular interference, similar to film grain. It is often seen in images acquired through ultrasound or synthetic aperture radar (SAR) imaging systems.

4.Quantization Noise: Quantization noise arises from the process of converting continuous analog signals into discrete digital values. It can cause slight variations in pixel intensities, particularly in areas of low contrast.

5.Motion Blur: Motion blur occurs when there is relative motion between the camera and the scene being photographed during the exposure. It results in a blurred appearance of moving objects in the image.

6.Lens Distortions: Lens distortions, such as chromatic aberration, vignetting, or geometric distortions, can introduce irregularities in the image due to imperfections in the camera optics.

4.2 Techniques and methods to remove noise:

Median Filtering:

Description: In median filtering, each pixel's value is replaced with the median value of its neighboring pixels within a specified window or kernel. This technique is particularly effective for removing salt-and-pepper noise, which appears as randomly occurring bright and dark pixels in the image.

Advantages: Median filtering is robust to outliers and preserves edges well, making it suitable for images affected by impulse noise.

Limitations: It may not be as effective for reducing other types of noise, such as Gaussian noise, and can cause blurring of fine details in the image.

Gaussian Filtering:

Description: Gaussian filtering involves convolving the image with a Gaussian kernel, which applies a weighted average to each pixel based on its neighbors' values. This process effectively smooths out noise in the image, particularly Gaussian noise, which is characterized by random fluctuations with a Gaussian distribution.

Advantages: Gaussian filtering preserves image details while reducing noise, making it suitable for applications where maintaining sharpness is important.

Limitations: It may blur edges in the image, leading to loss of detail in regions with high contrast transitions.

Bilateral Filtering:

Description: Bilateral filtering is a nonlinear filtering technique that takes into account both spatial and intensity differences between pixels when smoothing the image. It applies a weighted average to each pixel based on its spatial proximity and similarity in intensity to neighboring pixels.

Advantages: Bilateral filtering effectively reduces noise while preserving edges and fine details in the image, making it suitable for denoising tasks where maintaining image structure is crucial.

Limitations: Despite its effectiveness, bilateral filtering can be computationally expensive, especially for large kernels or high-resolution images. Additionally, it may not completely eliminate noise in regions with low contrast.

In summary, each of these image filtering techniques serves a specific purpose and is suited for different types of noise reduction. While median filtering is effective for removing impulse noise, Gaussian filtering is suitable for Gaussian noise reduction, and bilateral filtering excels at preserving image structure while reducing noise. The choice of filter depends on the characteristics of the noise present in the image and the desired balance between noise reduction and preservation of image details

4.3 Why is it important to remove background?

Removing the background from an image is important for several reasons, particularly in image processing and computer vision applications:

1.Object Detection and Recognition: Removing the background helps in isolating and highlighting the main objects or subjects present in the image. This makes it easier for object detection and recognition algorithms to focus on identifying the objects of interest without being distracted by irrelevant background elements.

2.Segmentation:
Background removal is often a preprocessing step in image segmentation, where the goal is to partition an image into meaningful regions or segments. By removing the background, the segmentation algorithm can focus solely on delineating the boundaries of objects within the foreground.

3.Visualization and Presentation:
Removing distracting backgrounds can improve the visual appeal of images, making them more aesthetically pleasing and easier to interpret. It also allows for better presentation of isolated objects without clutter.

4.Feature Extraction:
Background removal can facilitate the extraction of features from images for various analysis tasks. Features such as shape, texture, and color are often extracted from foreground objects, and removing the background helps in obtaining more accurate and meaningful feature representations.

5.Augmented Reality and Virtual Reality:
In applications involving augmented reality (AR) or virtual reality (VR), removing the background from images or video frames is crucial for seamlessly integrating virtual objects or overlays into the real-world scene.

6.Image Compression:
Background removal can reduce the complexity of images, making them more compressible without losing important information. This can lead to more efficient storage and transmission of images, particularly in applications where bandwidth or storage space is limited.

7.Improving Accuracy:
Background removal can improve the accuracy of various image analysis tasks, such as object classification, tracking, and counting. By isolating the foreground objects, the algorithms can focus on analyzing the relevant parts of the image more effectively.

Overall, removing the background from images is a fundamental step in many image processing and computer vision tasks, enabling better analysis, visualization, and interpretation of visual data. It helps in emphasizing the main objects or subjects in the image while reducing noise and distractions from the background.

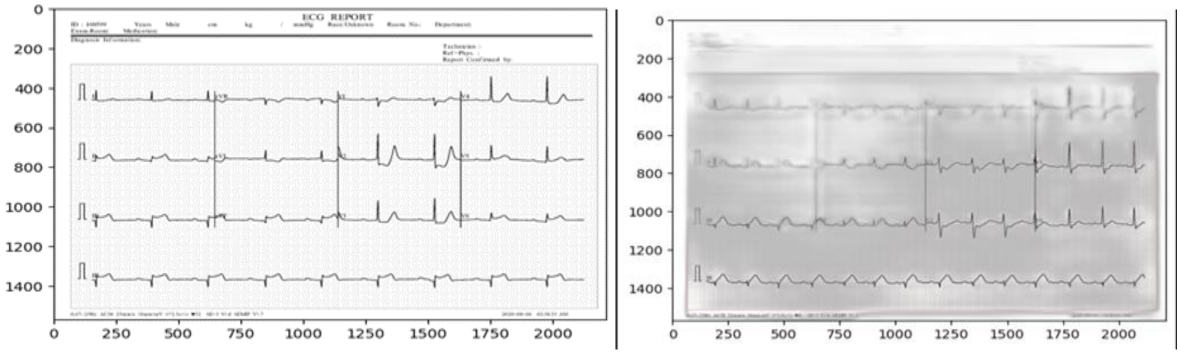
4.4 Techniques used to remove the background:

There are several types of technologies and methods used for removing backgrounds from images. Here are some of the main ones.

1.rembg :

Rembg is a tool that uses artificial intelligence techniques to remove background and Allows users to efficiently separate the backbone of an image from the background

But when it was used, it was a bad result.



before **after**

figure (5)

2.Traditional Image Processing Techniques:

Thresholding:

Segments the image based on a threshold value, separating foreground from background based on pixel intensity.

before **after**

figure (6)

3.Deep Learning-Based Methods:

•Mask R-CNN:

An extension of Faster R-CNN that also predicts segmentation masks for each detected object, allowing for precise object segmentation including background removal.

before **after**

figure (7)

Mask Technique:
Create a binary mask that indicates which pixels belong to the foreground (object of interest) and which pixels belong to the background. The mask can be generated manually, semi-automatically, or through automated segmentation techniques. Once obtained, the mask is applied to remove the background from the image.

Online Background Removal Tools:

Various online platforms offer automated background removal services where users can upload images, and the background is automatically removed using AI-powered algorithms.

Each technique has its advantages and limitations, and the choice depends on factors such as the complexity of the background, the presence of foreground objects, computational resources, and specific requirements of the application. Experimentation and evaluation are essential for selecting the most suitable background removal technique for a given scenario.

4.5 Why is important to make augmentation:

Data augmentation is crucial in machine learning and deep learning tasks, particularly in scenarios where training data is limited or when the model needs to generalize well to unseen data. Here are several reasons why data augmentation is important.

1.Increased Dataset Size:
Data augmentation effectively increases the size of the training dataset by generating new samples from existing ones. This helps in training more robust models and mitigates overfitting, especially when the original dataset is small.

2.Improved Model Generalization:
Augmenting the training data with variations such as rotations, translations, flips, and distortions exposes the model to a broader range of input variations. This encourages the model to learn more invariant and generalizable features, leading to better performance on unseen data.

3.Robustness to Input Variations:
By augmenting the data with various transformations, the model becomes more resilient to variations in input data, such as changes in lighting conditions,

viewpoints, or occlusions. This improves the model's ability to generalize across different real-world scenarios.

4.Regularization:

Data augmentation acts as a form of regularization by introducing noise or perturbations into the training data. This helps prevent the model from memorizing the training examples and encourages it to learn more robust and generalized representations.

5.Addressing Class Imbalance:

In classification tasks with imbalanced class distributions, data augmentation can be used to generate synthetic samples for minority classes, thereby balancing the class distribution and improving the model's ability to learn from underrepresented classes.

6.Reduced Annotation Costs:

Instead of manually collecting or annotating additional training data, data augmentation provides a cost-effective way to increase the diversity of the dataset. This is particularly beneficial in domains where data collection or annotation is expensive or time-consuming.

7.Domain Adaptation:

Data augmentation can help in adapting models trained on one domain to perform well in related but slightly different domains. By simulating variations in the target domain, data augmentation aids in transferring knowledge and adapting the model's representations.

4.6 Techniques and methods to make augmentation:

Here are various techniques and methods commonly used to perform data augmentation:

1.Geometric Transformations:

. **Rotation:** Rotate the image by a certain angle.
. **Translation:** Shift the image horizontally and/or vertically.
. **Scaling:** Enlarge or shrink the image.
. **Shearing:** Apply a shearing transformation to the image.

2.Flip and Mirror:

Horizontal Flip: Flip the image horizontally.
Vertical Flip: Flip the image vertically.

3.Crop and Pad:

- **Random Crop:** Randomly crop a portion of the image**.**
- **Center Crop:** Crop the center portion of the image**.**
- **Padding:** Add pixels around the edges of the image.

4.Color and Contrast Adjustments:

- **Brightness Adjustment:** Increase or decrease the brightness of the image.
- **Contrast Adjustment:** Increase or decrease the contrast of the image**.**
- **Saturation Adjustment:** Increase or decrease the saturation of colors in the image
- **Hue Adjustment:** Shift the hue of the image.

5.Blur and Sharpen:

•**Gaussian Blur:** Apply Gaussian blurring to the image.
•**Median Blur:** Apply median blurring to the image.
•**Sharpen:** Apply sharpening filters to enhance edges in the image.

6.Combining Transformations:

Apply a combination of different transformations to the image, such as rotation, scaling, and translation, sequentially or randomly

4.7 Scaling and normalization:

Scaling and normalization are both preprocessing techniques used to prepare data for machine learning models. While they share similarities, they serve different purposes.

- **Scaling:**

1.Purpose: Scaling aims to bring all feature values onto a similar scale, preventing features with larger magnitudes from dominating those with smaller magnitudes during model training.

2.Method: Scaling involves transforming feature values to a specific range, such as [0, 1] or [-1, 1]. Common scaling methods include min-max scaling and standardization (also known as z-score normalization).

3.Min-Max Scaling: Scales feature values to a specified range, often [0, 1], where the minimum value is mapped to 0, and the maximum value is mapped to 1. This method is useful when feature values have a clear minimum and maximum.

4.Standardization (Z-score Normalization): Rescales feature values to have a mean of 0 and a standard deviation of 1. This method assumes that the distribution of feature values is Gaussian (or nearly Gaussian). It is less affected by outliers compared to min-max scaling.

- **Normalization:**

1-Purpose: Normalization adjusts the values of features to a common scale, making it easier to compare the relative importance of different features.

2-Method: Normalization involves scaling each feature independently to have a mean of 0 and a standard deviation of 1. Unlike scaling, normalization does not change the shape of the feature distribution.

3-Normalization Formula: x'=std(x)/x−mean(x), where x is the original feature value' is the normalized feature value, mean(x) is the mean of the feature values, and std(x) is the standard deviation of the feature values.

In summary, scaling adjusts the range of feature values to a specific scale, while normalization standardizes feature values to have a mean of 0 and a standard deviation of 1. Both techniques are essential for ensuring that features contribute equally to the learning process and can improve the performance of machine learning models. The choice between scaling and normalization depends on the characteristics of the data and the requirements of the machine learning algorithm being used.

Chapter 05

Model Structure

5.1 Dataset Splitting:

The process of splitting a dataset into separate subsets for training, validation, and testing is a crucial step in building robust machine learning models. This section outlines the methodology used to divide the dataset, which consists of five distinct categories of ECG images: history of myocardial ECG, myocardial ECG, normal ECG, abnormal ECG, and COVID-19 ECG

Objective

The primary goal of dataset splitting is to ensure that the model is trained, validated, and tested on mutually exclusive subsets of data. This approach helps in evaluating the model’s performance accurately and prevents overfitting. In this project, we aim to allocate 70% of the data to the training set, 15% to the validation set, and the remaining 15% to the test set.

Methodology

Data Preparation:

- The dataset is stored in a directory structure where each subdirectory corresponds to one of the five ECG categories. Each subdirectory contains the respective images for that category.

Data Aggregation:

- We first aggregate all the image file paths and their corresponding labels into a pandas DataFrame. This DataFrame includes two columns: 'File_Path' and 'Label', where 'File_Path' contains the full path to each image file and 'Label' contains the integer-encoded category.

Label Conversion:

- To facilitate the compatibility with data generators and the training process, the integer labels are converted to string format.

Splitting Strategy:

- Using the train_test_split function from the sklearn.model_selection module, we split the DataFrame into three subsets:

- **Training Set (70%):** This subset is used to train the model. It includes 70% of the total dataset, ensuring that the model has sufficient data to learn the underlying patterns of each category.

- **Validation Set (15%):** This subset is used during the training process to validate the model's performance on unseen data. It helps in tuning the hyperparameters and selecting the best model. This set is obtained by splitting the remaining 30% of the data into two equal parts.

- **Test Set (15%):** This subset is used to evaluate the final model's performance. It is entirely independent of the training process and provides an unbiased assessment of how well the model generalizes to new data.

5.2 Utilized Model:

5.2.1 Inception 3:

Introduction

inception V3, a convolutional neural network (CNN) architecture, is part of the Inception family of models developed by researchers at Google. It is known for its efficiency in image classification tasks and is designed to handle the complexities of deep learning models without compromising computational efficiency. As an AI scientist, understanding the intricacies of Inception V3 is essential to leverage its full potential for various computer vision applications.

Applications

Inception V3's robust architecture and pre-training on the ImageNet dataset make it highly versatile for various computer vision applications, including:

1. **Image Classification:**
    - Inception V3 excels in categorizing images into a wide range of classes, making it ideal for tasks such as object recognition and scene understanding.
2. **Feature Extraction:**
    - The convolutional layers of Inception V3 can be used to extract meaningful features from images, which can then be fed into other models or used for tasks like image retrieval.
3. **Transfer Learning:**
    - Due to its pre-training on a large and diverse dataset, Inception V3 is well-suited for transfer learning. Fine-tuning the model on a new dataset can yield high performance with relatively few labeled examples.

4-**Medical Imaging:**

- Inception V3 has been applied in various medical imaging tasks, such as detecting abnormalities in ECG images, classifying skin lesions, and diagnosing diseases from radiology images.

5.2.1.1 Model Structure:

The Inception V3 architecture builds on the foundation laid by its predecessors, Inception V1 and V2, incorporating several key innovations and refinements that improve both accuracy and computational efficiency. Key components of Inception V3 include

1. Stem of the Network

The network starts with a few traditional convolution and pooling layers to reduce the spatial dimensions and prepare the input for the Inception modules.

- **Input Layer:** The network accepts an input image of size 299x299x3.
- **Convolution Layers:** The initial layers consist of standard convolutions with ReLU activations and batch normalization, which help to extract low-level features from the input image.
- **Max-Pooling:** A max-pooling layer is used to reduce the spatial dimensions and retain the most important features.
- **Factorized Convolutions:** 7x7 convolutions are factorized into smaller 3x3 convolutions to improve computational efficiency.
- **Pooling and Convolution Blocks:** Additional pooling and convolution operations further reduce the spatial dimensions while increasing the depth of the feature maps.

2. Inception Modules

Inception modules are the core building blocks of the Inception V3 architecture. These modules allow the network to capture multi-scale features by performing multiple convolutions with different filter sizes in parallel.

a. Inception Module A:

- Consists of several branches with 1x1, 3x3, and 5x5 convolutions.
- Incorporates factorized convolutional to improve efficiency.
- Ends with a concatenation of feature maps from all branches.

b. Inception Module B:

- Further factorizes convolutions into smaller, more efficient operations.
- Uses 1x7 and 7x1 convolutions to capture elongated features.
- Maintains multiple parallel branches for multi-scale feature extraction.

c. Inception Module C:

- Introduces more complex branching with different convolution and pooling operations.
- Uses 1x1, 3x3, and 1x3 convolutions.
- Combines outputs from all branches using concatenation.

3. Reduction Modules

Reduction modules are used to decrease the spatial dimensions of the feature maps significantly while preserving important features. There are two main reduction modules in Inception V3

a. Reduction Module A:

- Combines 3x3 max-pooling with 3x3 and 1x1 convolutions.
- Reduces the spatial dimensions and increases the depth of the feature maps.

b. Reduction Module B:

- Utilizes larger kernel sizes and more pooling operations.
- Provides a significant reduction in spatial dimensions to prepare the feature maps for the final layers.

4. Auxiliary Classifier

To combat the vanishing gradient problem in very deep networks, Inception V3 includes an auxiliary classifier connected to an intermediate layer. This classifier aids in training by providing additional gradient signals. It consists of

- A few convolutional layers.
- A fully connected layer.
- A softmax layer to output class probabilities.
- Used only during training to improve the gradient flow.

5. Final Layers

After passing through several Inception and reduction modules, the feature maps are processed by the final layers of the network:

- **Global Average Pooling (GAP):** Instead of using fully connected layers, Inception V3 employs global average pooling to reduce each feature map to a single number. This significantly reduces the number of parameters and helps prevent overfitting.
- **Fully Connected Layers:** A dense layer with 1024 units and ReLU activation.
- **Dropout:** A dropout layer with a dropout rate of 0.5 is used to further prevent overfitting.
- **Softmax Layer:** The final dense layer has as many units as the number of classes, with softmax activation to output the class probabilities.

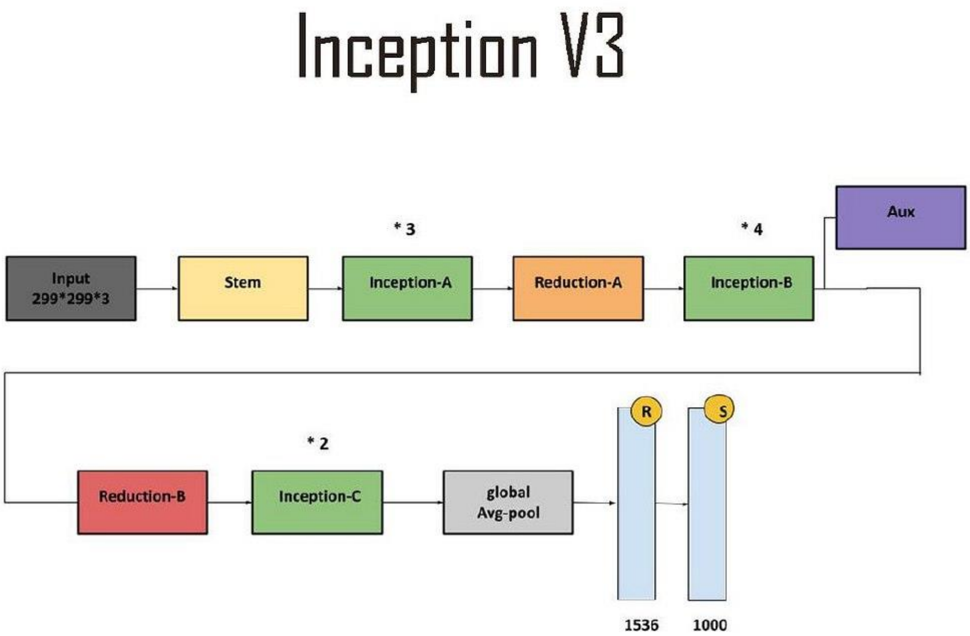


figure (8)

5.2.1.2training Parameters:

Training Parameters for Inception V3 Model

1. Learning Rate

- **Definition:** The learning rate controls the step size in the optimization process during training. It determines how quickly or slowly the model converges to the optimal solution**.**
- **Value Used:** 0.00001

- **Reason for Selection:** A lower learning rate was chosen to ensure gradual convergence, preventing the model from overshooting the optimal point. Additionally, the ReduceLROnPlateau callback was used to dynamically adjust the learning rate when the validation loss plateaued, reducing it by a factor of 0.2 with a minimum threshold of 1×10−61 \times 10^{-6}1×10−6.

2. Epochs

- **Definition:** An epoch is one complete pass through the entire training dataset. It defines how many times the learning algorithm will work through the entire training dataset.
- **Value Used:** 25
- **Reason for Selection:** The number of epochs was set to 25 to balance between adequate training and preventing overfitting. The EarlyStopping callback monitored the validation loss and halted training if no improvement was observed for 10 consecutive epochs, ensuring the model did not overfit.

3. Batch Size

- **Definition:** Batch size is the number of training samples used in one forward/backward pass. It influences the training speed and stability.
- **Value Used:** 32
- **Reason for Selection:** A batch size of 32 was chosen as it provides a good trade-off between computational efficiency and stable convergence.

4. Optimizer

- **Definition:** The optimizer updates the model parameters to minimize the loss function. It is crucial for the training process.
- **Optimizer Used:** Adam
- **Parameters:**
  - **Learning Rate:** 0.00001
  - **Beta_1:** 0.9 (exponential decay rate for the first moment estimates)
  - **Beta_2:** 0.999 (exponential decay rate for the second moment estimates)
  - **Epsilon:** 1×10−71 \times 10^{-7}1×10−7 (to prevent any division by zero in the implementation)
- **Reason for Selection:** Adam was chosen for its efficient and reliable convergence. It combines the advantages of both the AdaGrad and RMSProp algorithms, providing adaptive learning rates for each parameter.

5. Callbacks

- **Definition:** Callbacks are functions that are called at specific stages of the training process. They allow for dynamic adjustment of the training procedure.
- **Callbacks Used:**
  - **EarlyStopping:**

    Monitor: Validation loss

    Patience: 10 epochs

    Restore Best Weights: True

  - **ReduceLROnPlateau:**

    Monitor: Validation loss

    Factor: 0.2

    Patience: 5 epochs

    Min LR: 1×10−61 \times 10^{-6}1×10−6

- **Reason for Selection:**
  - EarlyStopping prevented overfitting by stopping the training when the validation loss did not improve for 10 consecutive epochs.
  - ReduceLROnPlateau dynamically adjusted the learning rate, reducing it by a factor of 0.2 when the validation loss plateaued, ensuring continued learning with finer adjustments.

6-Data Augmentation

- **Definition:** Data augmentation is a technique to increase the diversity of the training data without actually collecting new data. It involves creating modified versions of the images in the dataset.

- **Augmentation Techniques Used:**

  Scaling (1./255)

  Width shift range (0.2)

  Height shift range (0.2)

  Share range (0.2)

  Zoom range (0.2)

Horizontal flip

Fill mode (nearest)

- **Reason for Selection:** These augmentations were used to make the model more robust to various transformations and prevent overfitting by artificially increasing the diversity of the training set.

7. Model Architecture

- **Base Model:** InceptionV3 (pre-trained on ImageNet, without the top layer)
- **Custom Layers:**
  - **GlobalAveragePooling2D**
  - **Dense layer with 1024 units and ReLU activation**
  - **Dropout layer with a rate of 0.5**
  - **Dense layer with a number of units equal to the number of categories, with softmax activation for multi-class classification**

5.2.1.3 Evaluation & Result

Upon training the model using the specified parameters and datasets, we evaluated its performance on both the training and validation sets. Subsequently, we further assessed its generalization ability by testing it on an unseen test dataset.

Model Performance Metrics

- **Training Accuracy:** The model achieved a training accuracy of 97.59%, indicating its capability to learn and generalize patterns from the training data.
- **Validation Accuracy:** The validation accuracy reached 97.76%, suggesting that the model performed consistently well on data it hasn't been trained on, thereby validating its robustness.
- **Test Accuracy:** When evaluated on the unseen test dataset, the model demonstrated an accuracy of 97.20%, affirming its ability to generalize to new, previously unseen data.

These results highlight the effectiveness of the developed model in accurately classifying electrocardiogram (ECG) images into their respective categories. Such high accuracies across all datasets indicate the model's proficiency in learning meaningful features from the data and making accurate predictions.

5.2.1.4 Performance Metrics for Classification

In this section, we delve deeper into the performance metrics used to evaluate the classification model. These metrics provide a comprehensive understanding of the model's performance

Confusion Matrix

The confusion matrix allows us to visualize the performance of the classification model by showing the true positive, true negative, false positive, and false negative predictions across different classes.

Overall Confusion Matrix

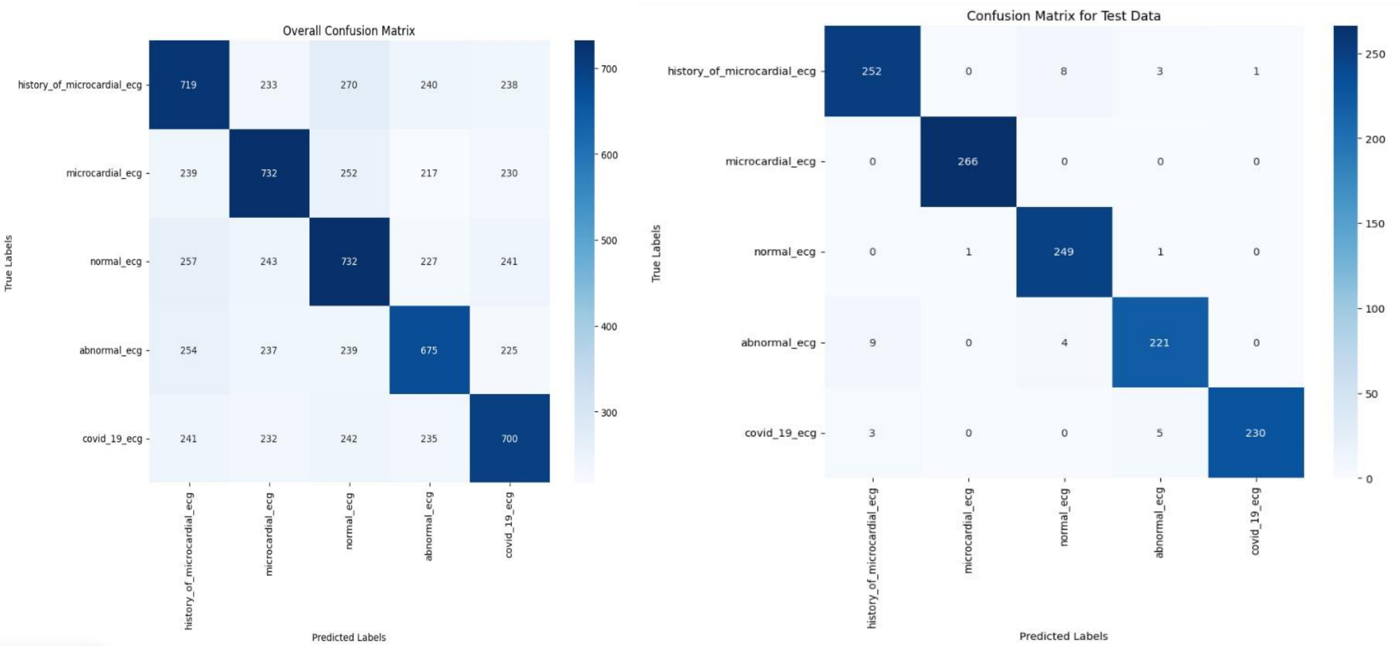
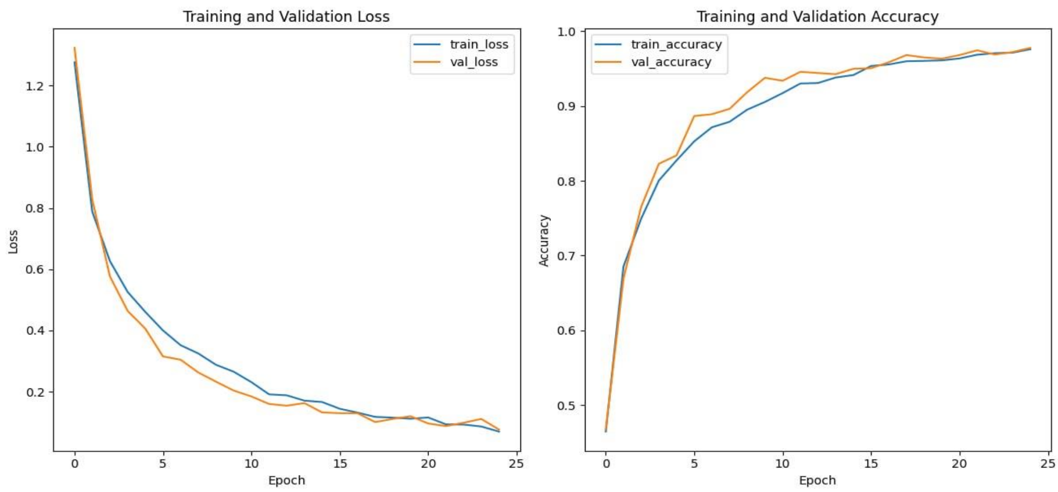


figure (9)

training and validation accuracy graph



5.3 Comparison of the performances of the different CNN models:

The first tried model is Resenet_50:

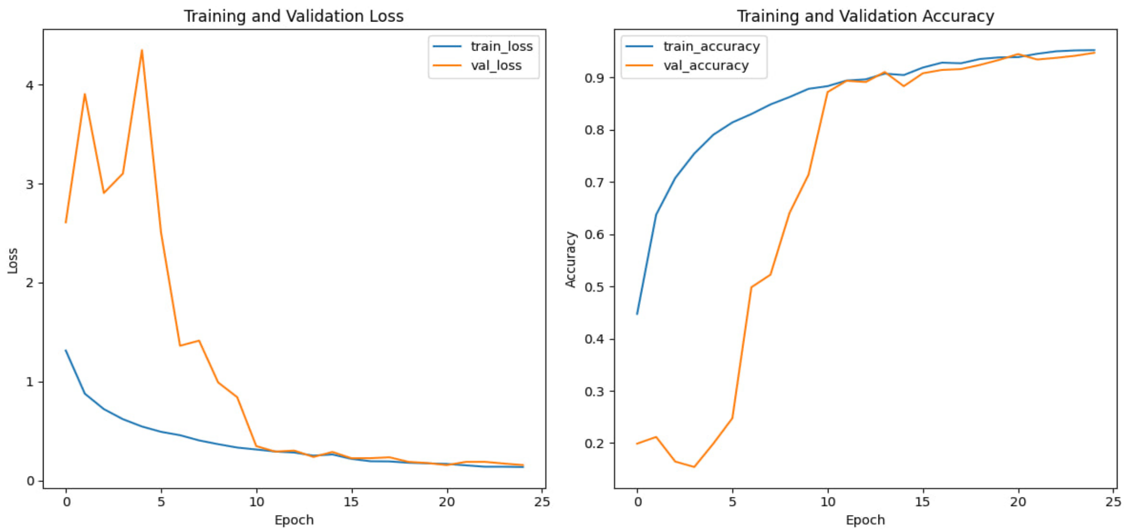
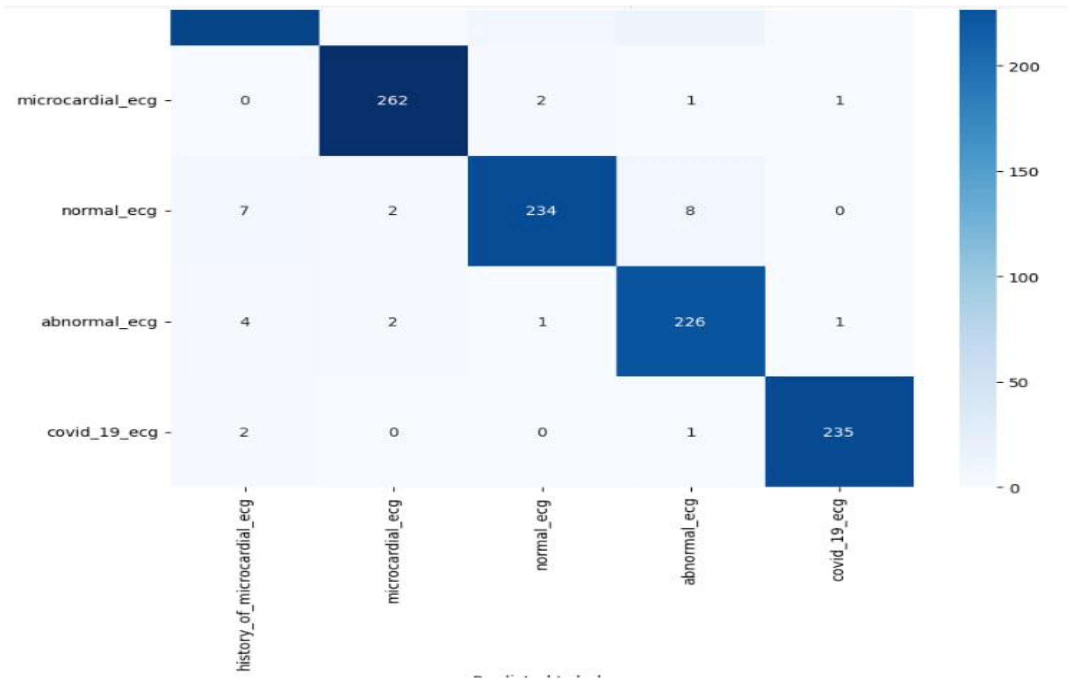


figure (10)



The second tried model is inception_v3:

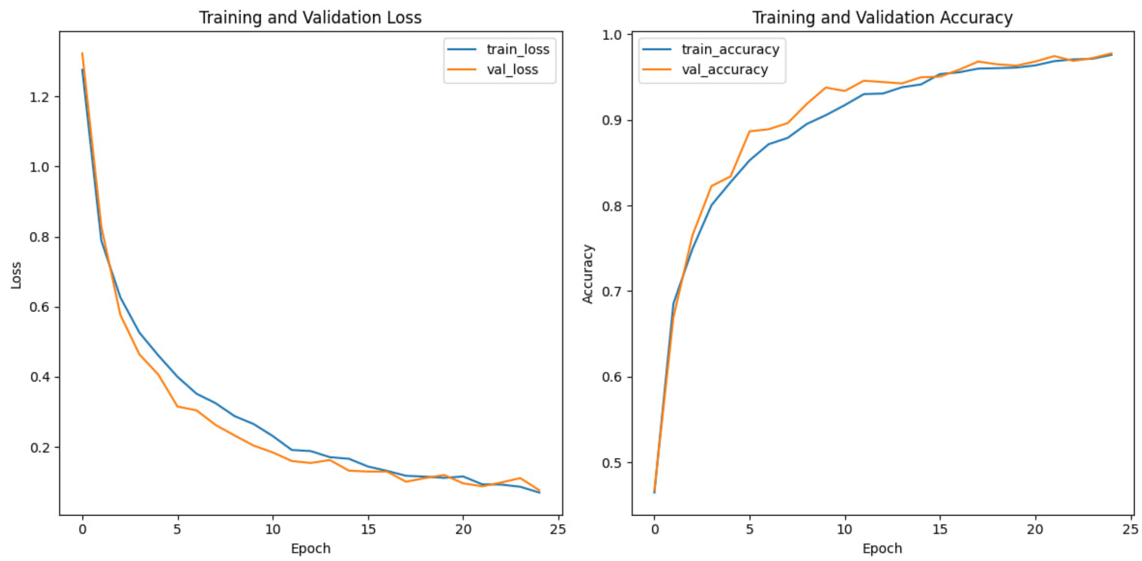
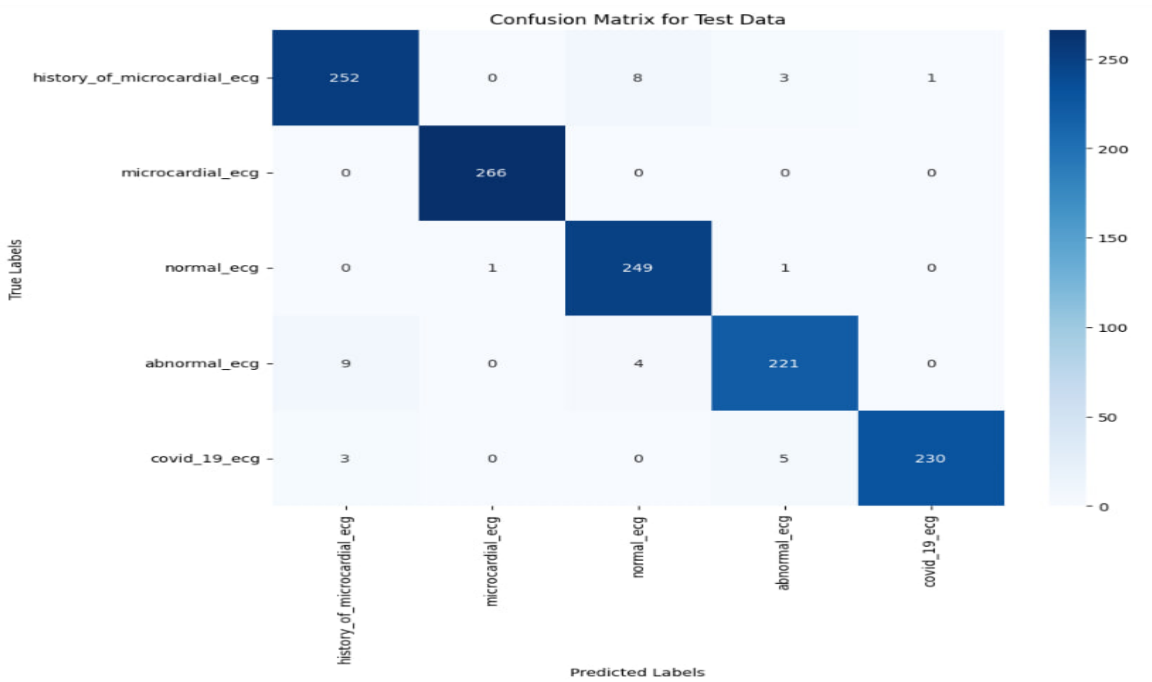


figure (11)



The third tried model is VGG_19:

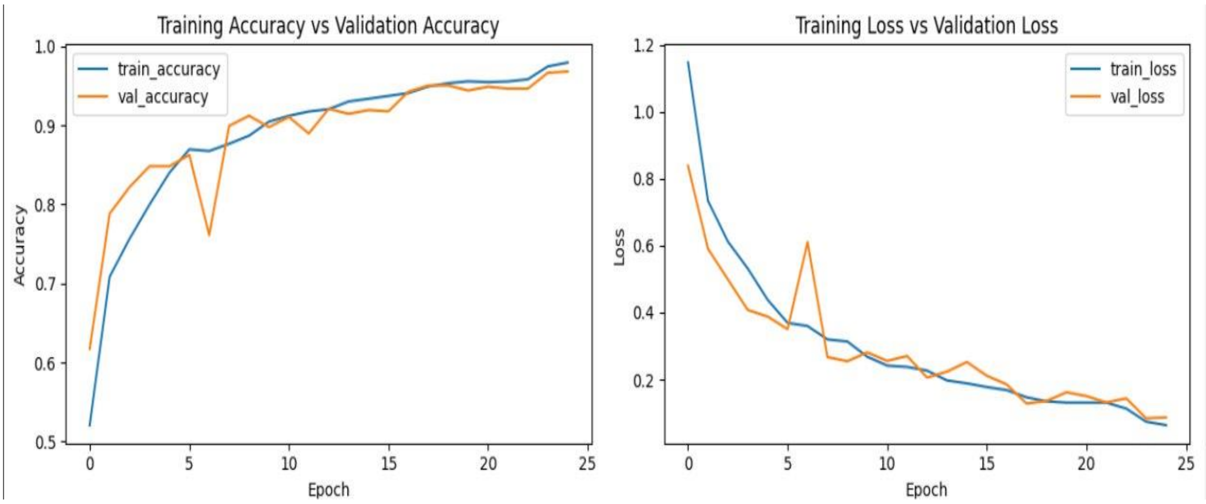
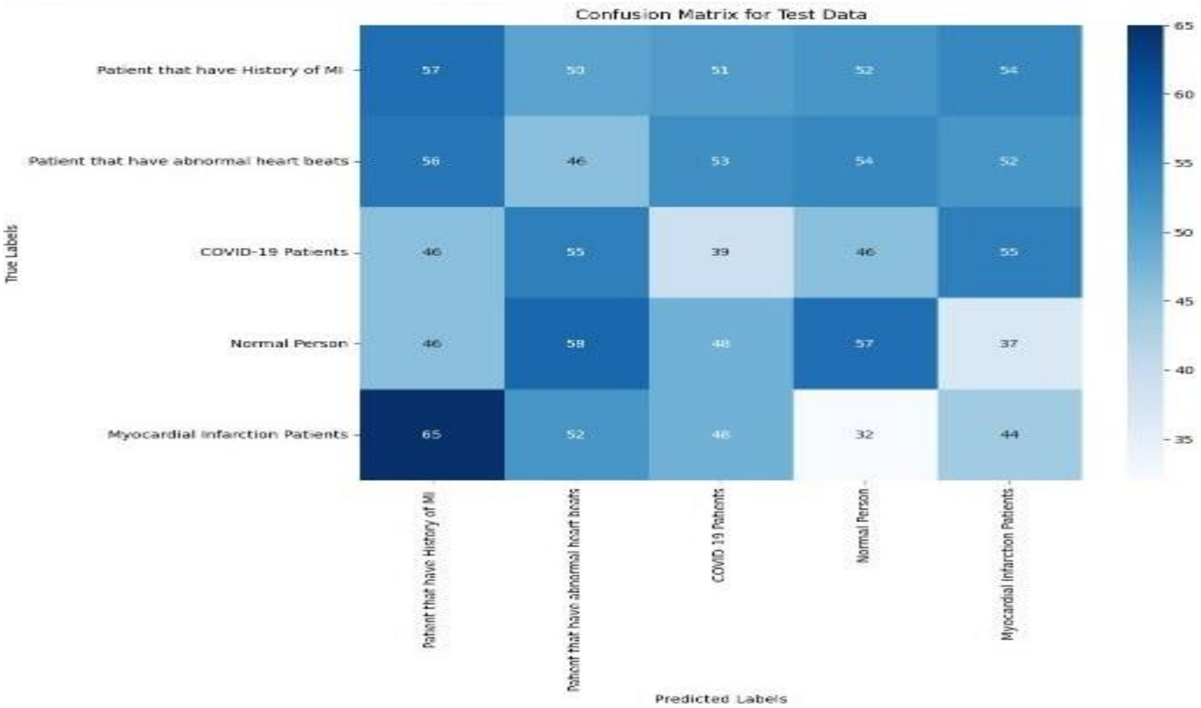


figure (12)



5.4 Implementation in code:

❖ **This code for all libraries import:**

```
import numpy as np

import pandas as pd

import os

from zipfile import*

import zipfile

import matplotlib.pyplot as plt

%matplotlib inline

from tqdm import tqdm

import tensorflow as tf

from tensorflow import keras

import cv2

from numpy import *

from keras.models import Sequential

from keras.layers import Dense,Flatten,MaxPooling2D,Convolution2D

from tensorflow.keras.preprocessing.image import
ImageDataGenerator, load_img, array_to_img, img_to_array

%matplotlib inline

import os



import cv2
```

- **This code for removing Noise from images:**

```
import cv2

import os



def process_image(input_directory, output_directory,
target_size=(1000, 1000)):

    os.makedirs(output_directory, exist_ok=True)



    for filename in os.listdir(input_directory):

        if filename.endswith((".jpg", ".jpeg", ".png")):

            input_image_path = os.path.join(input_directory,
filename)

            output_image_path = os.path.join(output_directory,
f"processed_{filename}")



            # Read the input image in grayscale

            img = cv2.imread(input_image_path,
cv2.IMREAD_GRAYSCALE)



            # Resize the image

            img_resized = cv2.resize(img, target_size)



            # Normalize the image

            #img_normalized = img_resized / 255.0  # Scale pixel
values to the range [0, 1]
```

```
            # Apply bilateral filtering

            bilateral_blur = cv2.bilateralFilter(img_resized,
d=9, sigmaColor=75, sigmaSpace=75)



            # Convert the image to 8-bit unsigned integer

            img_uint8 = bilateral_blur.astype('uint8')



            # Save the processed image

            cv2.imwrite(output_image_path, img_uint8)



            print(f"Processed image saved to:
{output_image_path}")



# Input and output directories

base_path = '/content/data2/ECG Images of COVID-19 Patients
(250)'

new_base_path = "/content/drive/MyDrive/covid_19_clean"  #
Corrected the output directory



# Target size for resizing

target_size = (1000, 1000)



# Process images with resizing, normalization, and bilateral blur

process_image(base_path, new_base_path, target_size)
```

❖ **This code for removing background from covid_19_images:**

```
import cv2
import numpy as np
import os

# Function to process an image and save the results
def process_and_save(input_path, output_folder):
    # Load the image
    img = cv2.imread(input_path)

    # Check if the image is successfully loaded
    if img is None:
        print(f"Error: Image not loaded. Check the file
path: {input_path}")
        return

    # Convert the color space
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Define the lower and upper white color thresholds
    lower_white = np.array([200, 200, 200], dtype=np.uint8)
    upper_white = np.array([255, 255, 255], dtype=np.uint8)

    # Create a mask based on the white color thresholds
    mask = cv2.inRange(img, lower_white, upper_white)

    # Perform morphological operations on the mask
    mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3)))

    # Invert the mask
    mask = cv2.bitwise_not(mask)

    # Create a background with white color
    bk = np.full(img.shape, 255 ,dtype=np.uint8)
```

```
    # Apply the mask to the original image
    fg_masked = cv2.bitwise_and(img, img, mask=mask)

    # Invert the mask again
    mask = cv2.bitwise_not(mask)

    # Apply the inverted mask to the background
    bk_masked = cv2.bitwise_and(bk, bk, mask=mask)

    # Combine the masked foreground and background
    final = cv2.bitwise_or(fg_masked, bk_masked)

    # Invert the mask one more time
    mask = cv2.bitwise_not(mask)

    # Specify the output file path for saving
    output_path = os.path.join(output_folder,
os.path.basename(input_path))

    # Save the final image in the output folder
    cv2.imwrite(output_path, final)

# Input folder path
input_folder = "/content/drive/MyDrive/covid_19_clean"

# Output folder path
output_folder =
"/content/drive/MyDrive/covid_19_clean_1back"

# Create the output folder if it doesn't exist
os.makedirs(output_folder, exist_ok=True)

# Process all images in the input folder
for img_name in os.listdir(input_folder):
    img_path = os.path.join(input_folder, img_name)
    process_and_save(img_path, output_folder)

print(f"Images processed and saved successfully in the
folder: {output_folder}")
```

❖ **This code for removing background from all other categories:**

```
import cv2
import numpy as np
import os

# Function to process an image and save the results
def process_and_save(input_path, output_folder):
    # Load the image
    img = cv2.imread(input_path)

    # Check if the image is successfully loaded
    if img is None:
        print(f"Error: Image not loaded. Check the file
path: {input_path}")
        return

    # Convert the color space
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

    # Define the lower and upper white color thresholds
    lower_white = np.array([170, 170, 170], dtype=np.uint8)
    upper_white = np.array([255, 255, 255], dtype=np.uint8)

    # Create a mask based on the white color thresholds
    mask = cv2.inRange(img, lower_white, upper_white)

    # Perform morphological operations on the mask
    mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN,
cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3)))

    # Invert the mask
    mask = cv2.bitwise_not(mask)

    # Create a background with white color
    bk = np.full(img.shape, 255 ,dtype=np.uint8)
```

```
    # Apply the mask to the original image
    fg_masked = cv2.bitwise_and(img, img, mask=mask)

    # Invert the mask again
    mask = cv2.bitwise_not(mask)

    # Apply the inverted mask to the background
    bk_masked = cv2.bitwise_and(bk, bk, mask=mask)

    # Combine the masked foreground and background
    final = cv2.bitwise_or(fg_masked, bk_masked)

    # Invert the mask one more time
    mask = cv2.bitwise_not(mask)

    # Specify the output file path for saving
    output_path = os.path.join(output_folder,
os.path.basename(input_path))

    # Save the final image in the output folder
    cv2.imwrite(output_path, final)

# Input folder path
input_folder = "/content/drive/MyDrive/ab_clean"

# Output folder path
output_folder ="/content/drive/MyDrive/ab_clean_BACK"

# Create the output folder if it doesn't exist
os.makedirs(output_folder, exist_ok=True)

# Process all images in the input folder
for img_name in os.listdir(input_folder):
    img_path = os.path.join(input_folder, img_name)
    process_and_save(img_path, output_folder)

print(f"Images processed and saved successfully in the
folder: {output_folder}")
```

❖ **This code for removing header and footer from covid_19_images:**

```
import cv2
import os

def remove_header_footer(image_path, output_folder,
header_percentage=10, footer_percentage=10):
    # Load the image
    img = cv2.imread(image_path)

    # Calculate the number of rows to be removed from the
top and bottom
    height, width, _ = img.shape
    header_rows = int((header_percentage / 100) * height)
    footer_rows = int((footer_percentage / 170) * height)

    # Remove rows from the top and bottom
    img_cropped = img[header_rows:-footer_rows, :]

    # Specify the output file path for saving
    output_path = os.path.join(output_folder,
os.path.basename(image_path))

    # Save the result
    cv2.imwrite(output_path, img_cropped)

# Input folder path
input_folder = "/content/drive/MyDrive/covid_19_clean_1back"

# Output folder path
output_folder =
"/content/drive/MyDrive/covid_19_clean_remove"

# Create the output folder if it doesn't exist
os.makedirs(output_folder, exist_ok=True)

# Process all images in the input folder
```

```
for img_name in os.listdir(input_folder):
    img_path = os.path.join(input_folder, img_name)
    remove_header_footer(img_path, output_folder,
header_percentage=10, footer_percentage=10)
```

❖ **This code for removing header and footer from other categories of images:**

```
import cv2
import os

def remove_header_footer(image_path, output_folder,
header_percentage=10, footer_percentage=10):
    # Load the image
    img = cv2.imread(image_path)

    # Calculate the number of rows to be removed from the
top and bottom
    height, width, _ = img.shape
    header_rows = int((header_percentage / 50) * height)
    footer_rows = int((footer_percentage / 170) * height)

    # Remove rows from the top and bottom
    img_cropped = img[header_rows:-footer_rows, :]

    # Specify the output file path for saving
    output_path = os.path.join(output_folder,
os.path.basename(image_path))

    # Save the result
    cv2.imwrite(output_path, img_cropped)

# Input folder path
input_folder = "/content/drive/MyDrive/ab_clean_BACK"

# Output folder path
output_folder =
"/content/drive/MyDrive/ab_clean_BACK_remove"
```

```
# Create the output folder if it doesn't exist
os.makedirs(output_folder, exist_ok=True)

# Process all images in the input folder
for img_name in os.listdir(input_folder):
    img_path = os.path.join(input_folder, img_name)
    remove_header_footer(img_path, output_folder,
header_percentage=10, footer_percentage=10)
```

❖ **This code for making augmentation from all categories of images:**

```
from keras.preprocessing.image import ImageDataGenerator,
array_to_img, img_to_array, load_img
import os

# Input and output directories
input_directory =
"/content/drive/MyDrive/covid_19_clean_remove"
output_directory = "/content/drive/MyDrive/covid_19_aug"

# Create the output directory if it doesn't exist
os.makedirs(output_directory, exist_ok=True)

# Create an ImageDataGenerator
datagen = ImageDataGenerator(rotation_range=7,
                             horizontal_flip=True,
                             vertical_flip=True,
                             channel_shift_range=4,
                             shear_range=0.06,  # Shear
transformation
                             width_shift_range=0.085,  #
Horizontal shift
                             height_shift_range=0.085)  #
Vertical shift)  # Fill mode for points outside the input
boundaries

# Process each image in the input directory
for filename in os.listdir(input_directory):
    if filename.endswith((".jpg", ".jpeg", ".png")):
```

```
        input_image_path = os.path.join(input_directory,
filename)

        # Load the image
        img = load_img(input_image_path)
        x = img_to_array(img)
        x = x.reshape((1,) + x.shape)

        # Generate 10 augmented images and save them
        i = 0
        for batch in datagen.flow(x, batch_size=1,
save_to_dir=output_directory,

save_prefix=f"{os.path.splitext(filename)[0]}_aug",
                                 save_format='jpeg'):
            i += 1
            if i >= 6:  # Generate 10 augmented images per
original image
                break

print("Data augmentation complete.")
```

❖ **This code for making mode of inception_v3:**

```
import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.layers import Dense,
GlobalAveragePooling2D, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import
ImageDataGenerator
from tensorflow.keras.callbacks import EarlyStopping,
ReduceLROnPlateau
import pandas as pd
import joblib
```

```
# Define data folder and categories
data_folder = '/content/data2_after_aug'
categories = ['Patient have History of MI', 'Patient have
abnormal heart beats', 'COVID-19 Patients', 'Normal Person',
'Myocardial Infarction Patients']

# Define image parameters
image_size = (299, 299)  # Default size for InceptionV3
batch_size = 32

# Create DataFrame with file paths and corresponding labels
file_paths = []
labels = []
for i, category in enumerate(categories):
    category_folder = os.path.join(data_folder, category)
    category_files = [os.path.join(category_folder, file)
for file in os.listdir(category_folder)]
    file_paths.extend(category_files)
    labels.extend([i] * len(category_files))

data_df = pd.DataFrame({'File_Path': file_paths, 'Label':
labels})

# Convert integer labels to string labels
data_df['Label'] = data_df['Label'].astype(str)

# Split data into train, validation, and test sets
train_df, test_val_df = train_test_split(data_df,
test_size=0.3, random_state=42)
val_df, test_df = train_test_split(test_val_df,
test_size=0.5, random_state=42)  # 0.5 x 0.3 = 0.15 for
validation and test each

# Data generators with augmentation
datagen = ImageDataGenerator(
    rescale=1./255,
    width_shift_range=0.2,
    height_shift_range=0.2,
```

```
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

train_generator = datagen.flow_from_dataframe(
    train_df,
    x_col='File_Path',
    y_col='Label',
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

validation_generator = datagen.flow_from_dataframe(
    val_df,
    x_col='File_Path',
    y_col='Label',
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)

test_generator = datagen.flow_from_dataframe(
    test_df,
    x_col='File_Path',
    y_col='Label',
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)

# Load the InceptionV3 pre-trained model
```

```python
base_model = InceptionV3(weights='imagenet',
include_top=False, input_shape=(image_size[0],
image_size[1], 3))

# Add custom classification layers with dropout
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)  # Add dropout to prevent overfitting
predictions = Dense(len(categories),
activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

# Unfreeze some layers of the base model for fine-tuning
for layer in base_model.layers[-30:]:
    layer.trainable = True

# Compile the model with a lower learning rate
model.compile(optimizer=Adam(learning_rate=0.00001),
loss='categorical_crossentropy', metrics=['accuracy'])

# Callbacks for early stopping and learning rate reduction
early_stopping = EarlyStopping(monitor='val_loss',
patience=10, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss',
factor=0.2, patience=5, min_lr=1e-6)

# Train the model with callbacks
history = model.fit(
    train_generator,
    epochs=25,
    validation_data=validation_generator,
    callbacks=[early_stopping, reduce_lr]
)

# Evaluate model on test data
test_loss, test_accuracy = model.evaluate(test_generator)
print(f"Test Loss: {test_loss}")
```

```
print(f"Test Accuracy: {test_accuracy}")

# Plot training history including loss and accuracy
plt.figure(figsize=(12, 6))

# Plot training and validation loss
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='train_loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.title('Training and Validation Loss')
plt.legend()

# Plot training and validation accuracy
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'],
label='train_accuracy')
plt.plot(history.history['val_accuracy'],
label='val_accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend()

plt.tight_layout()
plt.show()

# Save the model in h5 format
model.save("/content/drive/MyDrive/trained_model5_inceptionv
3old.h5")

# Save the model in joblib format
joblib_file =
"/content/drive/MyDrive/trained_model5_inceptionv3old.joblib
"
joblib.dump(model, joblib_file)
```

- **This code for making confusion matrix:**

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Predict labels for test data
test_generator.reset()  # Reset the generator
test_predictions =
model.predict(test_generator).argmax(axis=-1)

# Get true labels for test data
test_true_labels = test_generator.labels

# Create confusion matrix for test data
test_conf_matrix = confusion_matrix(test_true_labels,
test_predictions)

# Plot the confusion matrix
plt.figure(figsize=(10, 8))
sns.heatmap(test_conf_matrix, annot=True, fmt="d",
cmap="Blues", xticklabels=categories,
yticklabels=categories)
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.title('Confusion Matrix for Test Data')
plt.show()

# Optionally, print the classification report for more
detailed metrics
from sklearn.metrics import classification_report
print("Classification Report for Test Data:")
print(classification_report(test_true_labels,
test_predictions, target_names=categories))
```

Chapter 06

Methodology

6.1 Structure System:

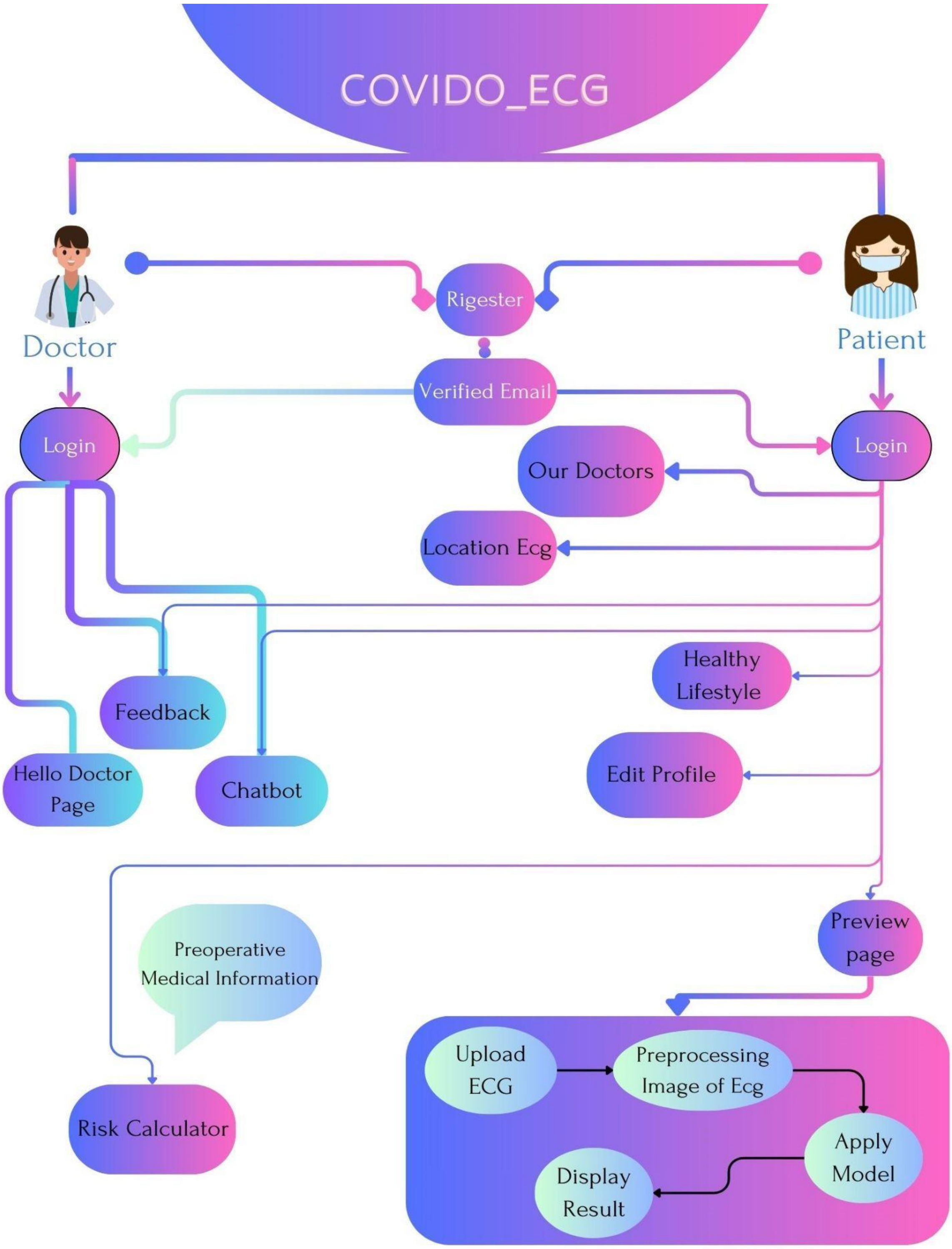


figure (13)

6.2 use cases:

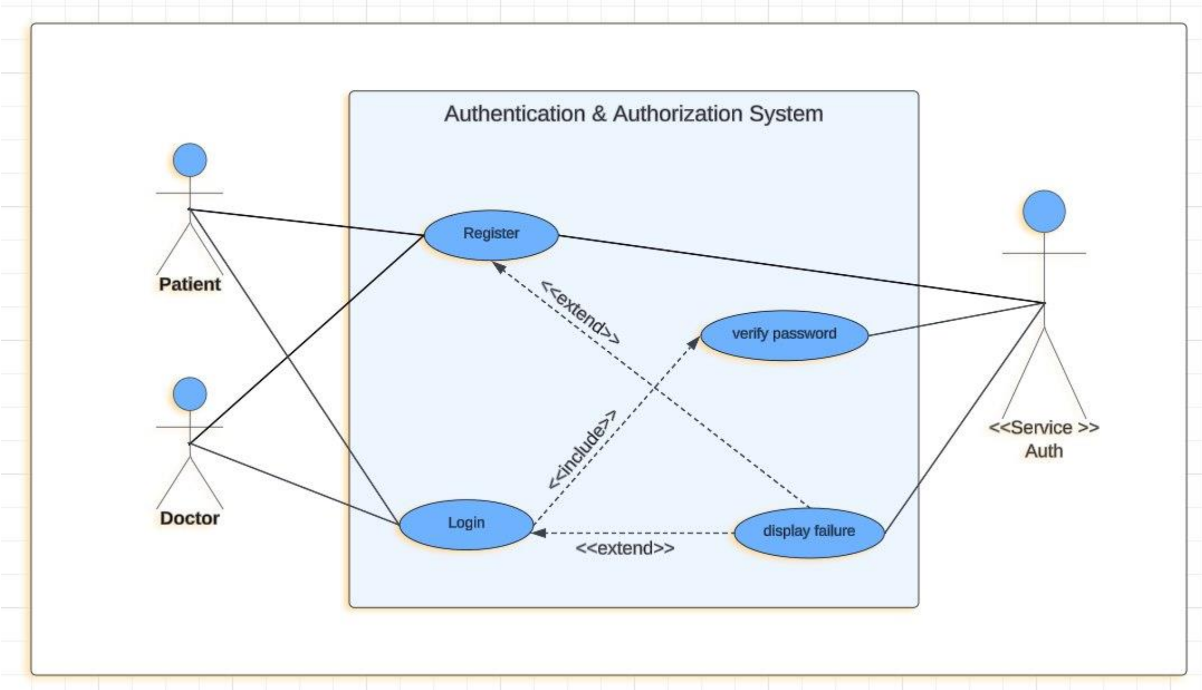


figure (14)

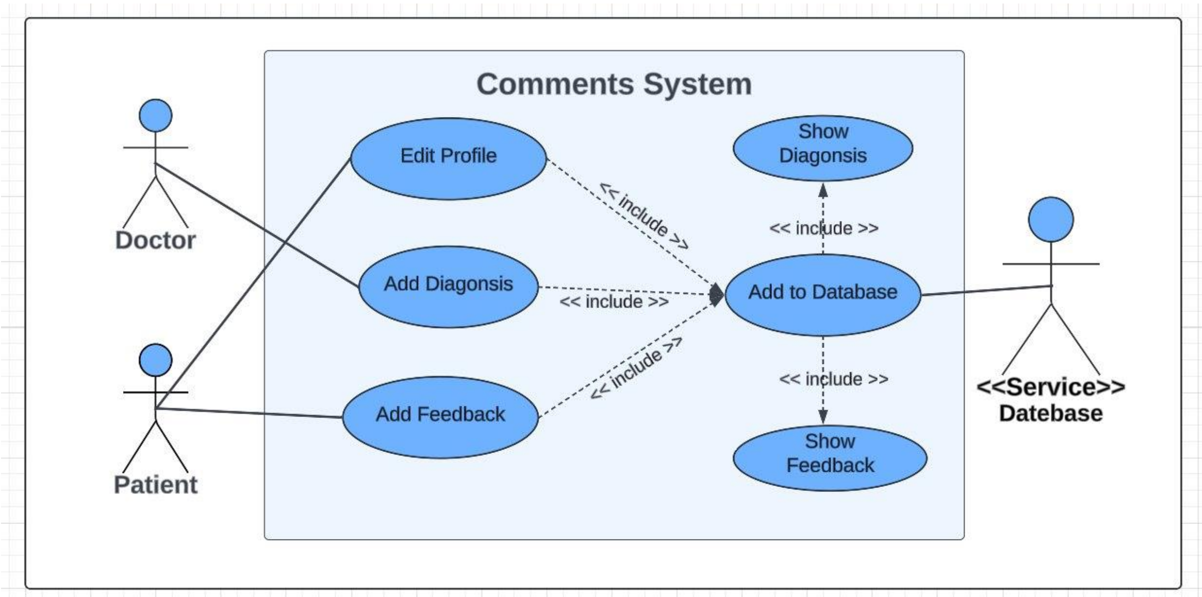


figure (15)

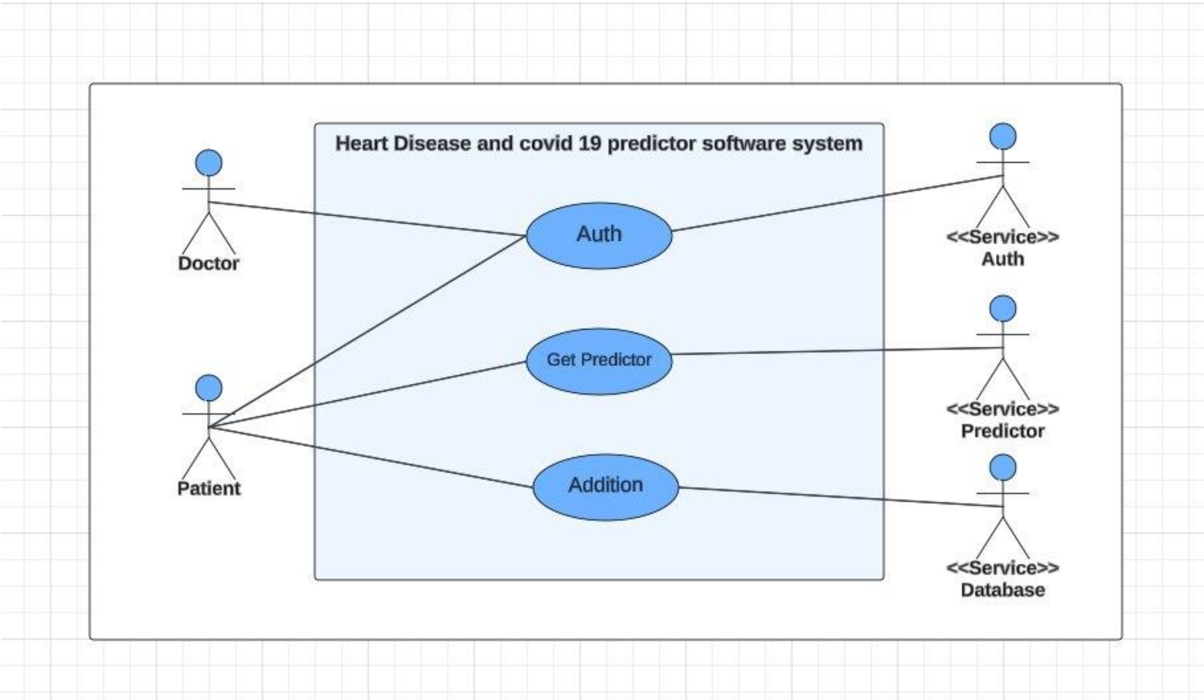


Figure (16)

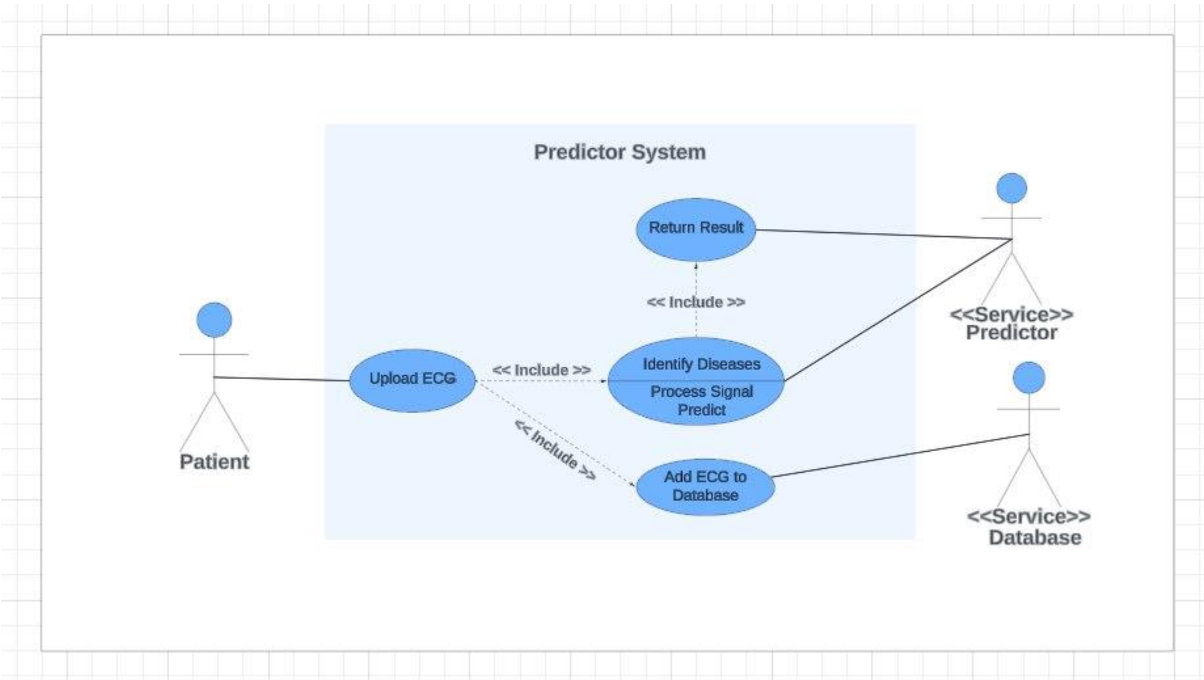


figure (17)

6.3 class diagram:

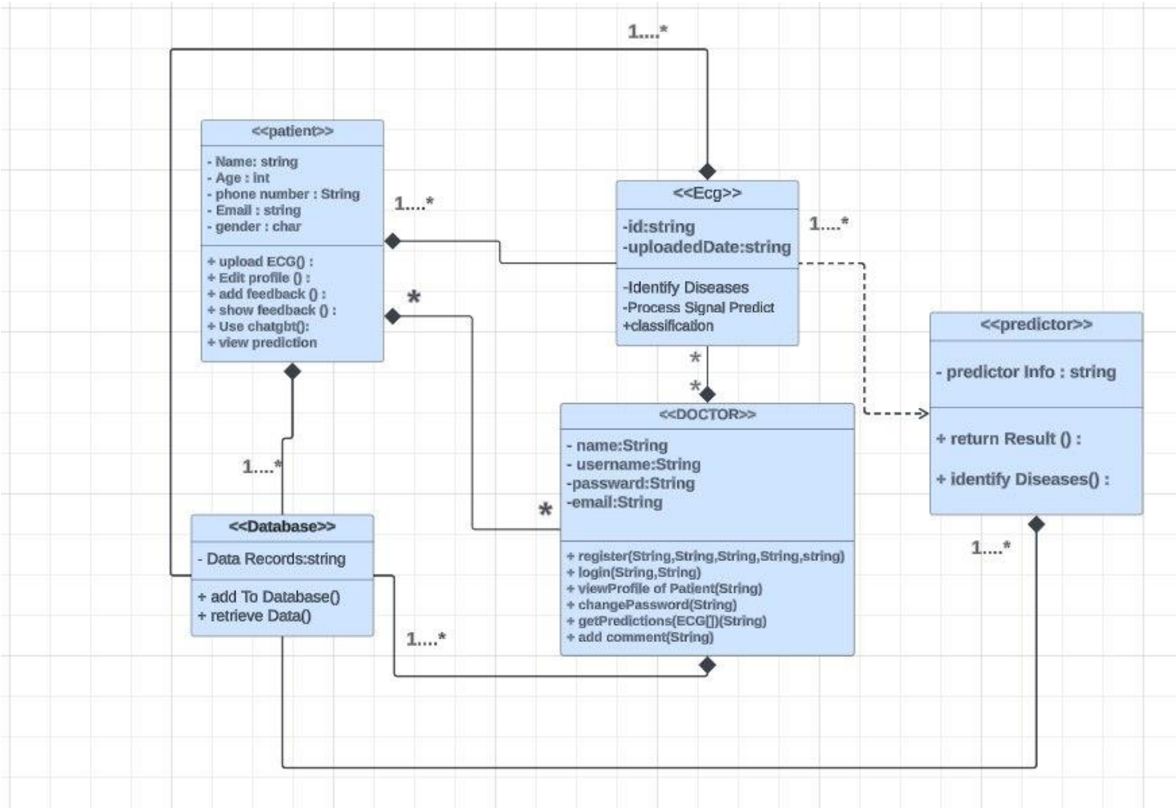


figure (18)

6.4 sequence diagram:

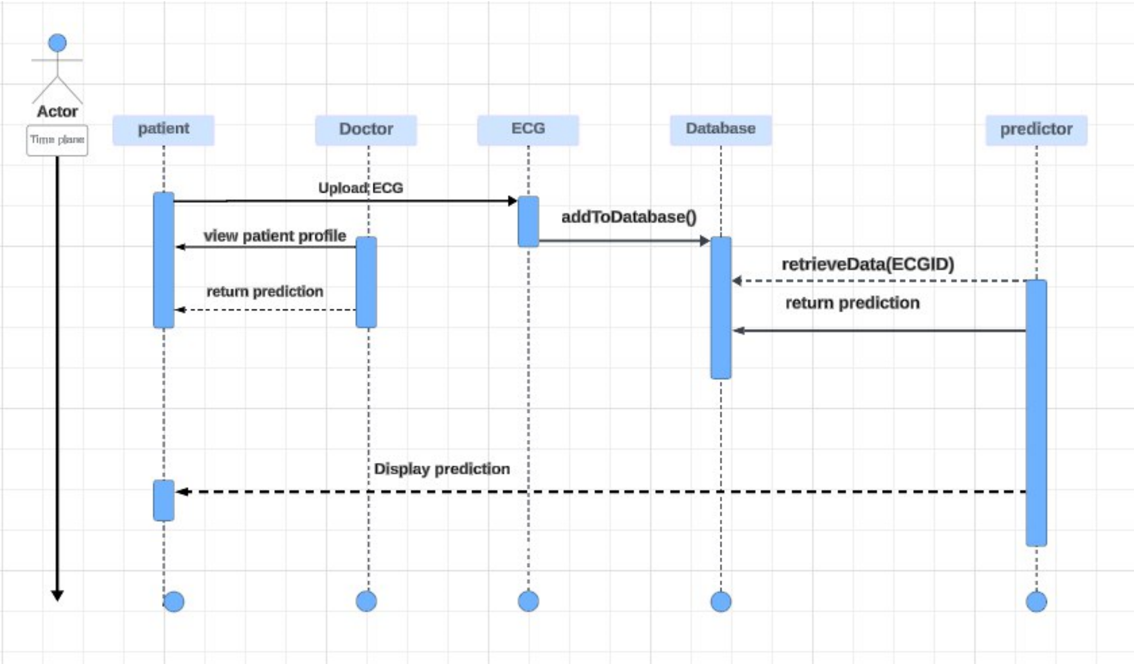


figure (19)

Chapter 07

Implementation

7.1 User Interface

7.1.1 Home Page:

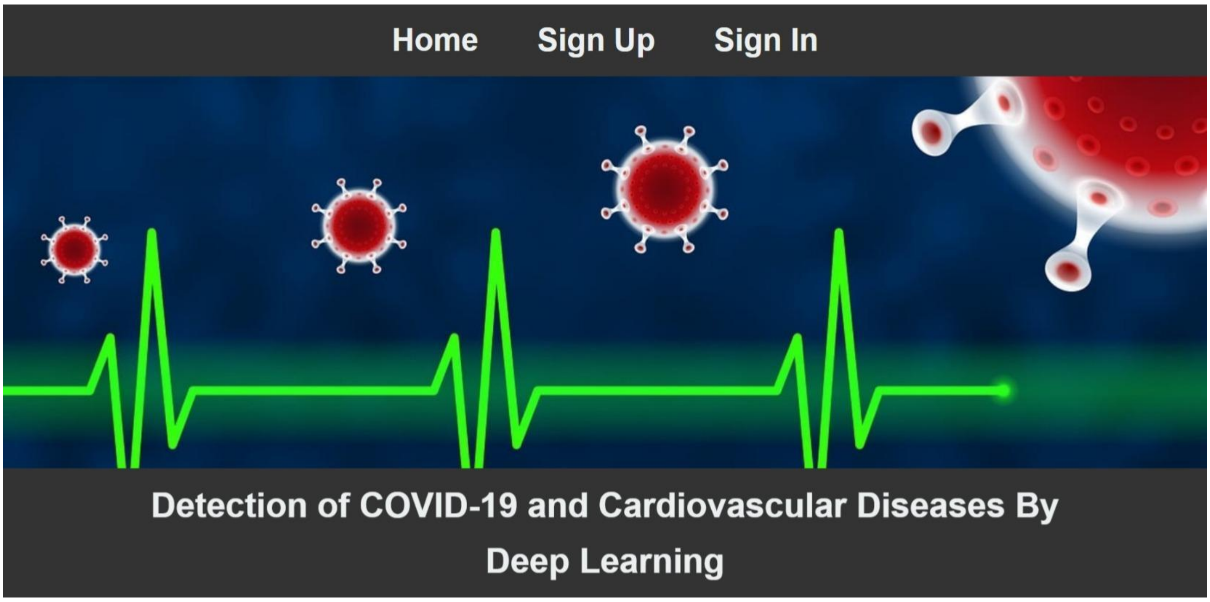


figure (20)

The home page is the first page that appears when you open a specific app on your phone or visit a website. It typically includes elements such as:

1. Main Menus: Providing quick access to different sections of the app or website.
2. Quick Links: Allowing fast navigation to important pages or sections within the app or website.

The main goal of the home page is to provide users with a comfortable and organized experience, making it easy for them to quickly and easily access the content they are looking for.

7.1.2 Sign Up Page:

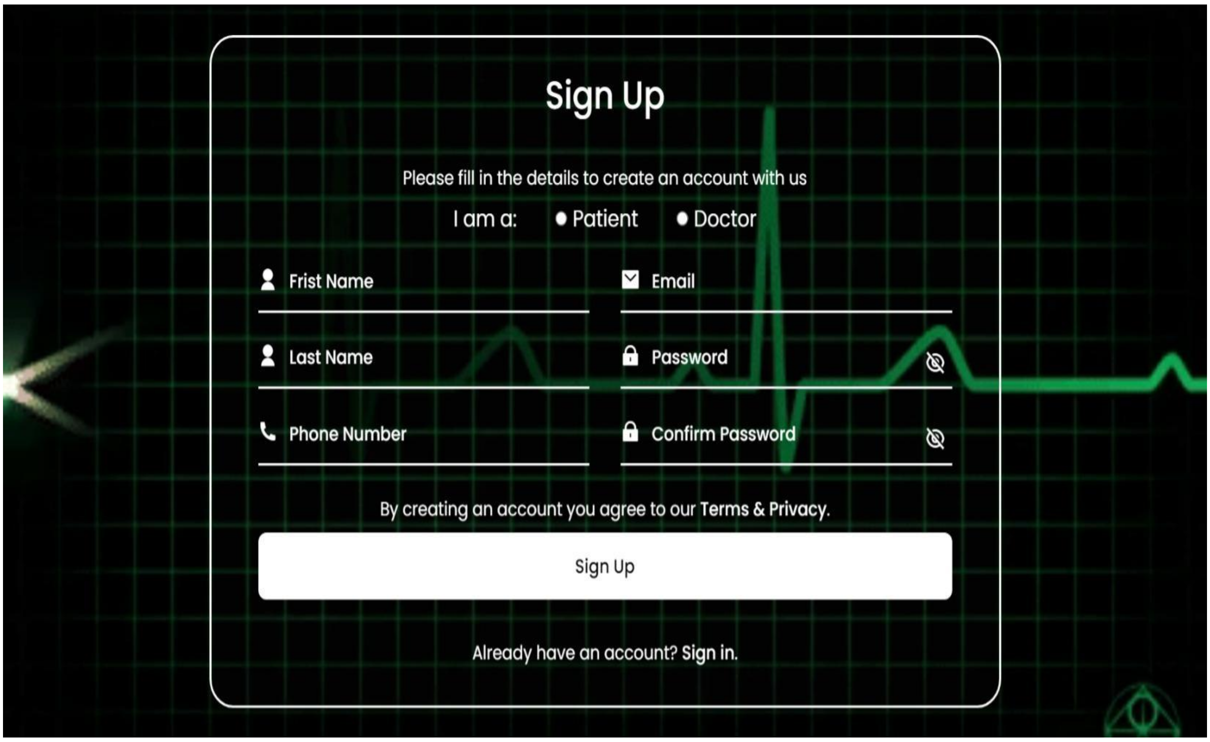


figure (21)

A sign-up page, also known as a registration page, is a web page or a user interface element that allows new users to create an account or register for a website, application, or online service.

The main purposes of a sign-up page are:

1. User Acquisition:

-It provides a way for potential users to join and become part of the product's user base.

-It helps to grow the user population and expand the customer reach**.**

2. Data Collection:

-It gathers essential user information such as name, email address, and password.

-This data is typically stored in a user database and used for various

purposes, such as account management, communication

3. Account Creation:

-It facilitates the creation of a new user account, which grants the user access to the product's features and functionalities.

-The account information allows users to log in and maintain their profile within the system.

4. User Onboarding:

-It guides new users through the registration process, providing instructions and feedback to ensure a smooth onboarding experience.

-This helps to reduce friction and increase the likelihood of users successfully completing the sign-up flow.

5. Security and Privacy:

-It often includes security measures, such as password requirements, to protect user accounts and prevent automated registrations.

-It may also display information about the product's privacy policy and terms of service to establish trust and transparency with the users.

The design and implementation of a sign-up page are crucial for the overall user experience and the success of user acquisition strategies. An effective sign-up page should balance simplicity, ease of use, and security to encourage users to complete the registration process.

7.1.3 Sign In Page:

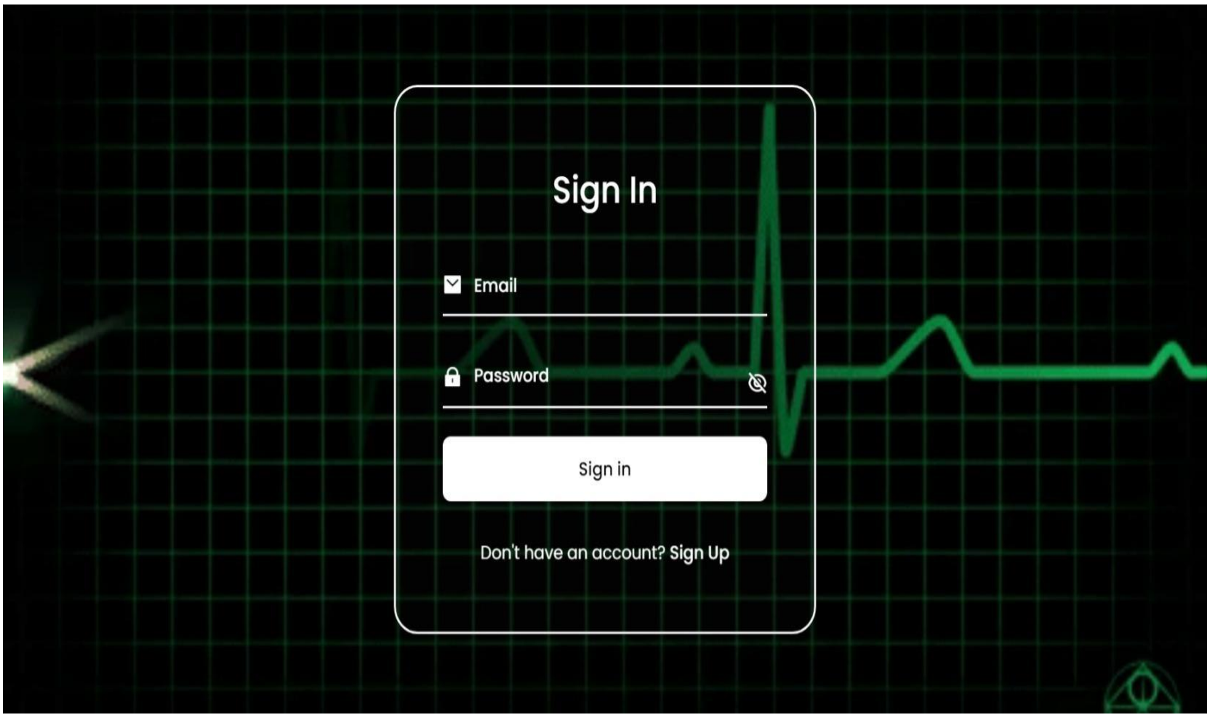


figure (22)

The Sign-in Page is a web page that allows registered users to log into their accounts.

Simply put, when this page is opened, the user will find two places to enter:

1. Username: This is the name or email address that the user registered when creating the account.

2. Password: This is the secret password that the user uses to verify their identity.

After entering this information, the system verifies it, and if it is correct, it allows the user to log in to their account and access the available features and services.

So in general, the sign-in page is the way registered users log in to their accounts on a particular website or application.

7.1.4 Preview Page:

In this page, the user uploads an Electrocardiogram (ECG) image for the results to be displayed. The results indicate whether the person is infected with a heart disease or COVID-19 or not.

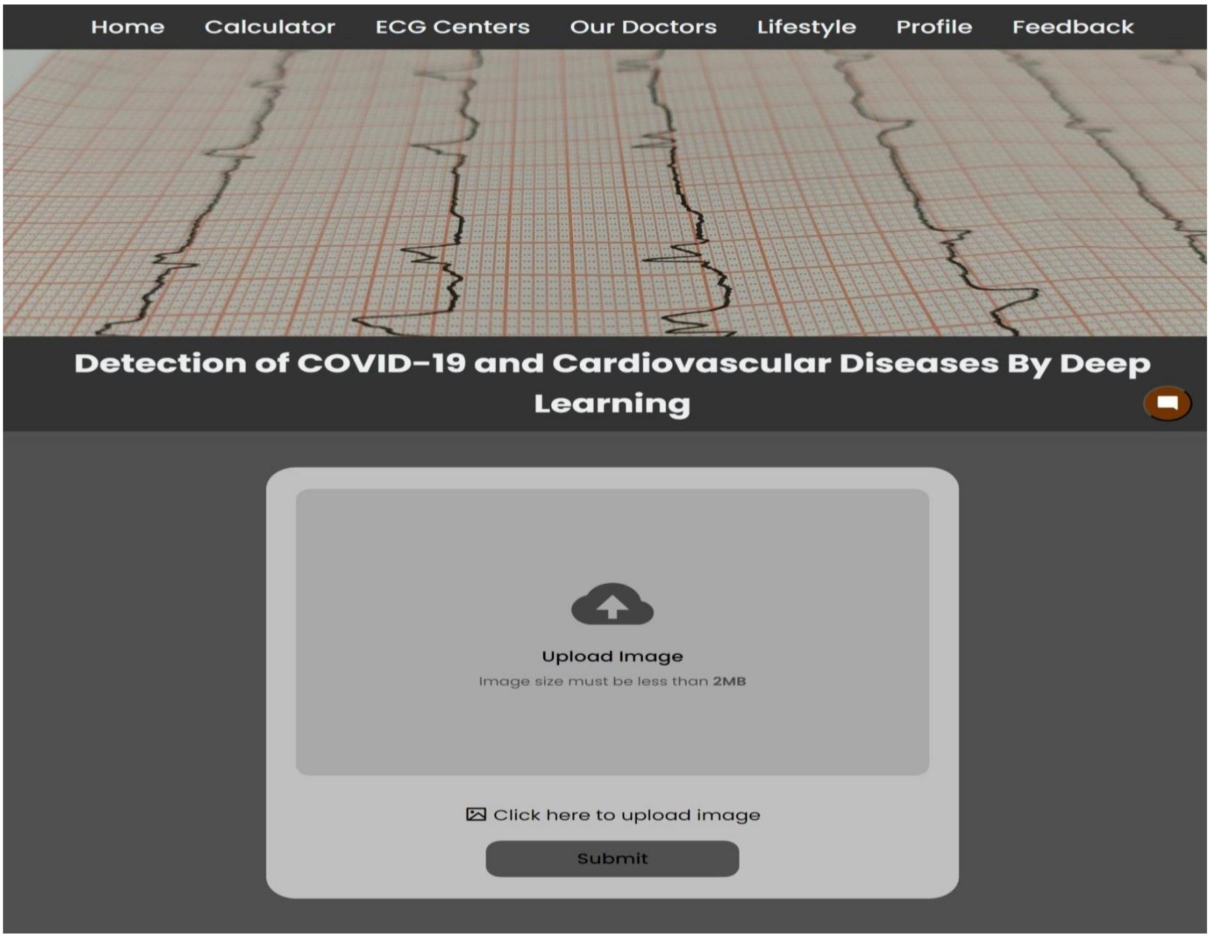


figure (23)

7.1.5 Calculator Page:

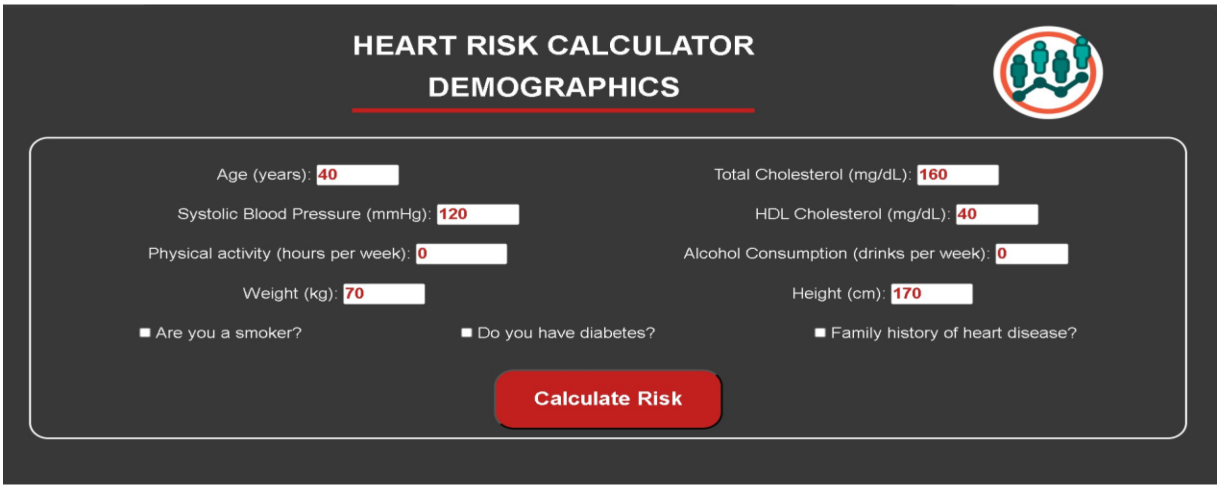


figure (24)

A tool designed to assess a user's risk of developing diseases, particularly cardiovascular diseases, based on their health and lifestyle data.

Input Data:

The Risk Calculator requires the following data from the user to evaluate their risk:

1. Age (years): The user's age, as age is a significant risk factor for many diseases.

2. Total Cholesterol (mg/dL): The total cholesterol level in the user's blood, which is a key indicator of heart disease risk.

3.Systolic Blood Pressure (mmHg): The user's systolic blood pressure, indicating the force exerted by blood on artery walls during heartbeats**.**

4. HDL Cholesterol (mg/dL): The level of high-density lipoprotein (HDL) cholesterol, known as "good" cholesterol, which helps reduce heart disease risk.

5. Physical Activity (hours per week): The amount of time the user spends on physical activities each week, influencing overall health.

6. Alcohol Consumption (drinks per week): The number of alcoholic drinks the user consumes weekly, as excessive alcohol can increase health risks.

7. Weight (kg): The user's weight, used to calculate body mass index (BMI) and assess overall health.

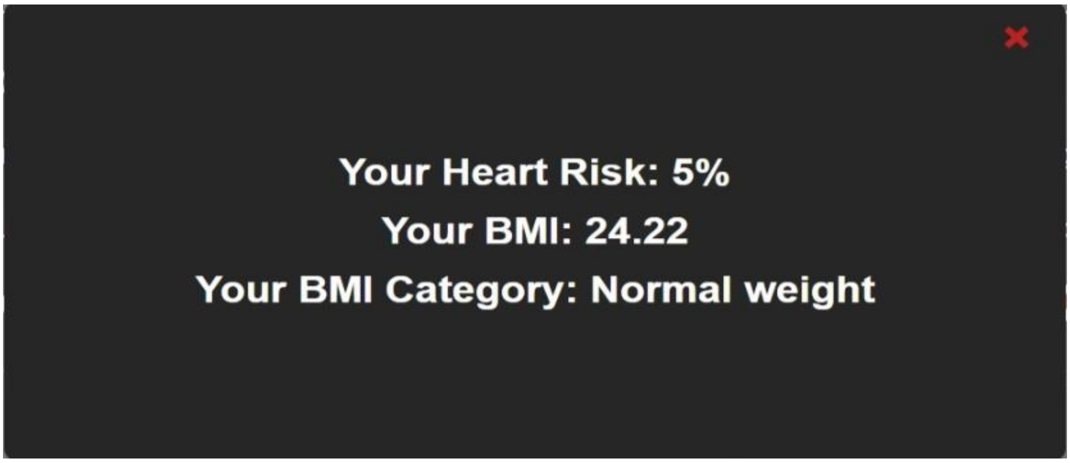
8.Height (cm): The user's height, also used in calculating BMI.

9.Smoker: Whether the user smokes, as smoking is a major risk factor for many diseases.

10.Diabetes: Whether the user has diabetes, which significantly raises the risk of cardiovascular diseases.

11-Family History of Heart Disease: Whether the user has a family history of heart disease, as genetics can influence disease risk.

The Result:



7.1.6 Patient Profile Page:

This page contains private information for each patient:

Personal Information: Name, Age, Email address, Phone number

Health-Related Information: Height, Weight, Total cholesterol, HDL cholesterol, Systolic blood pressure, Diastolic blood pressure, Physical Activity, Alcohol Consumption.

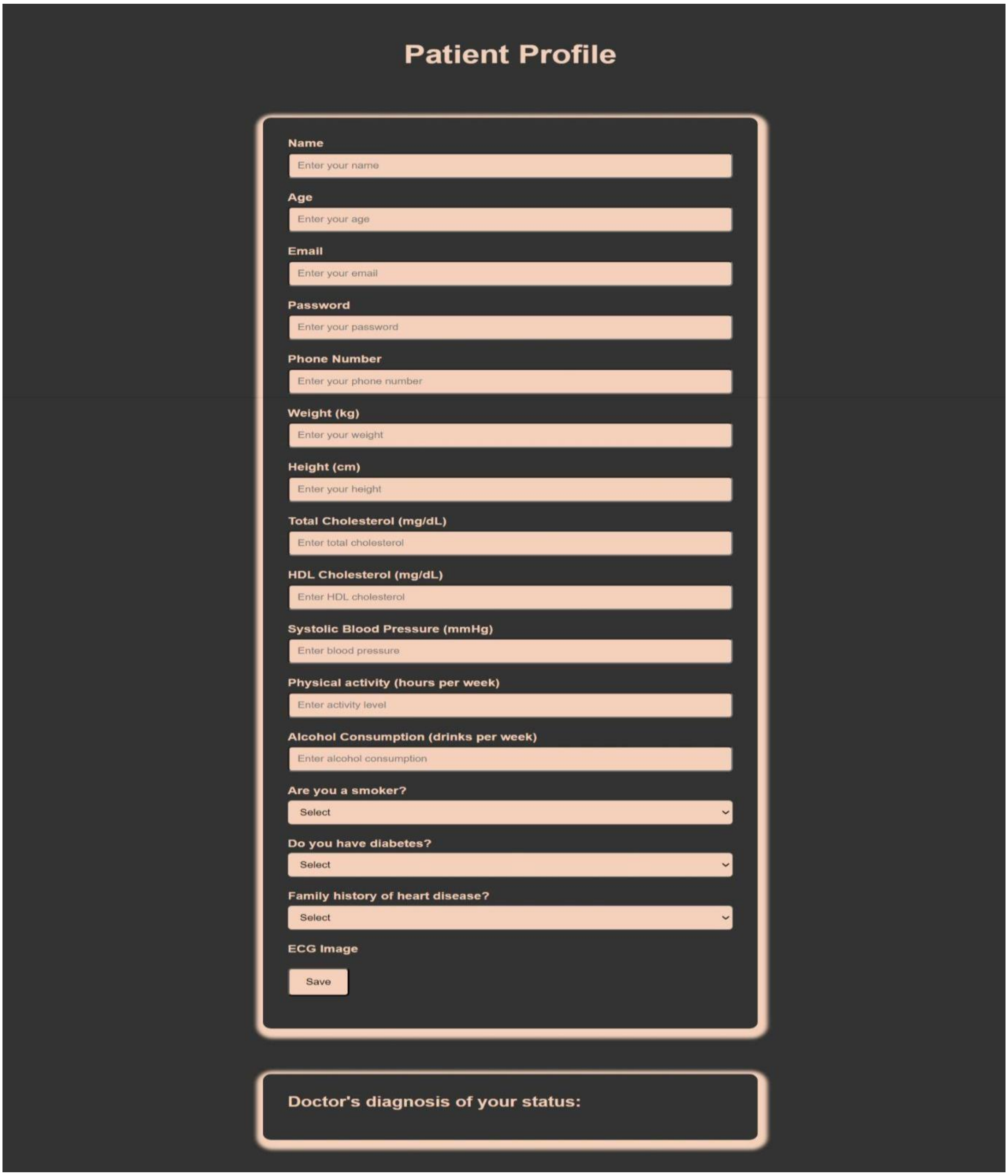

Patient Profile
Name
Enter your name
Age
Enter your age
Email
Enter your email
Password
Enter your password
Phone Number
Enter your phone number
Weight (kg)
Enter your weight
Height (cm)
Enter your height
Total Cholesterol (mg/dL)
Enter total cholesterol
HDL Cholesterol (mg/dL)
Enter HDL cholesterol
Systolic Blood Pressure (mmHg)
Enter blood pressure
Physical activity (hours per week)
Enter activity level
Alcohol Consumption (drinks per week)
Enter alcohol consumption
Are you a smoker?
Select
Do you have diabetes?
Select
Family history of heart disease?
Select
ECG Image
Save
Doctor's diagnosis of your status:

figure (25)

7.1.7 Doctor Diagnosis Page:

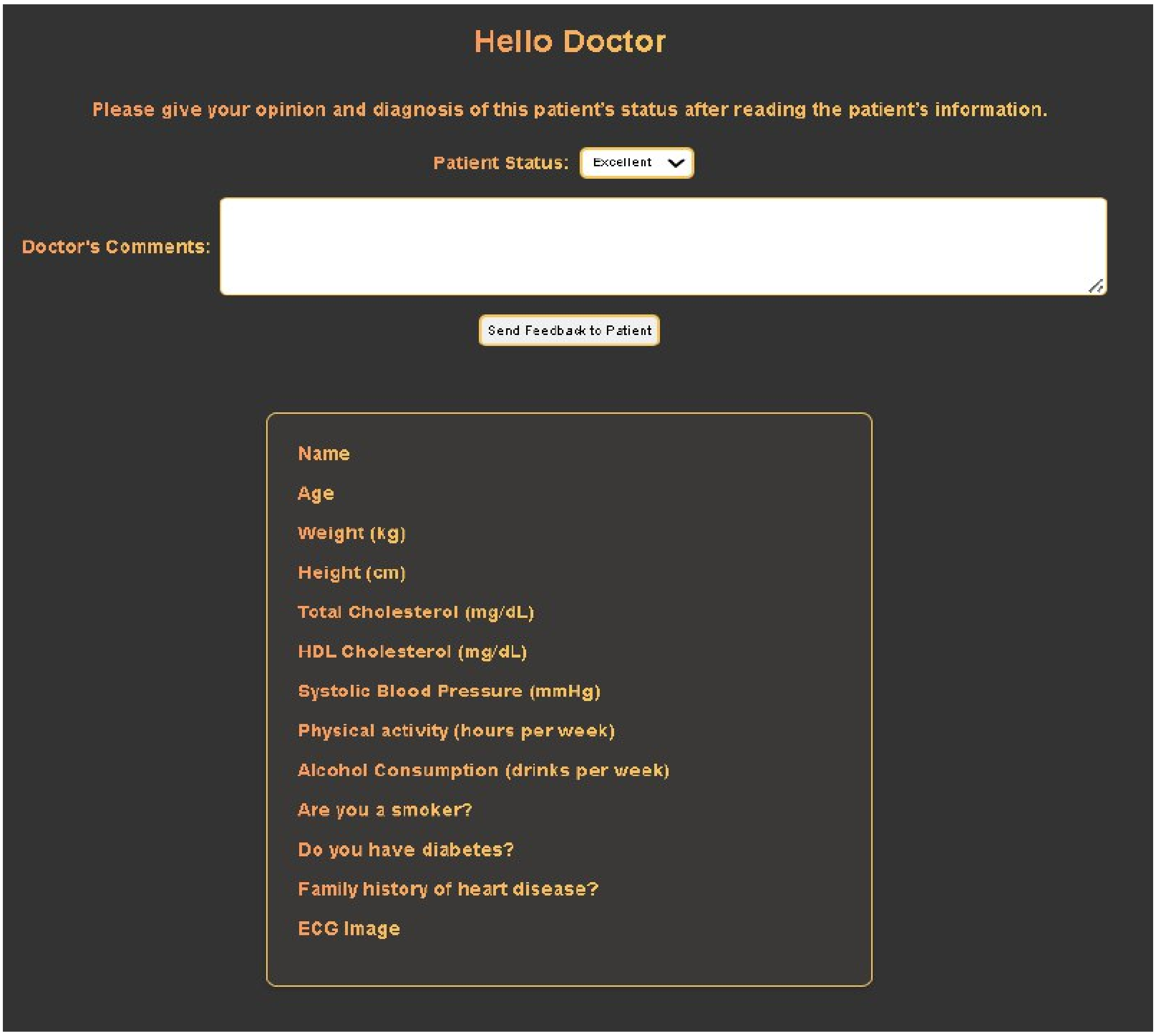


figure (26)

In this page, the doctor examines the patient's condition to determine if it is good or not. After that, the doctor provides comments for the patients.

1.Guide the patient: The doctor can guide the patient about the next steps in the treatment process and clarify future expectations.

2.Enhance communication: This page helps enhance communication and interaction between the doctor and the patient through providing the feedback.

3.Monitor the case: The doctor can follow up on the patient's condition and progress in treatment through this page.

- The main goal of this page is to improve the quality of healthcare provided to patients by enhancing communication and giving direct feedback from the doctor.

7.1.8 Chatbot:

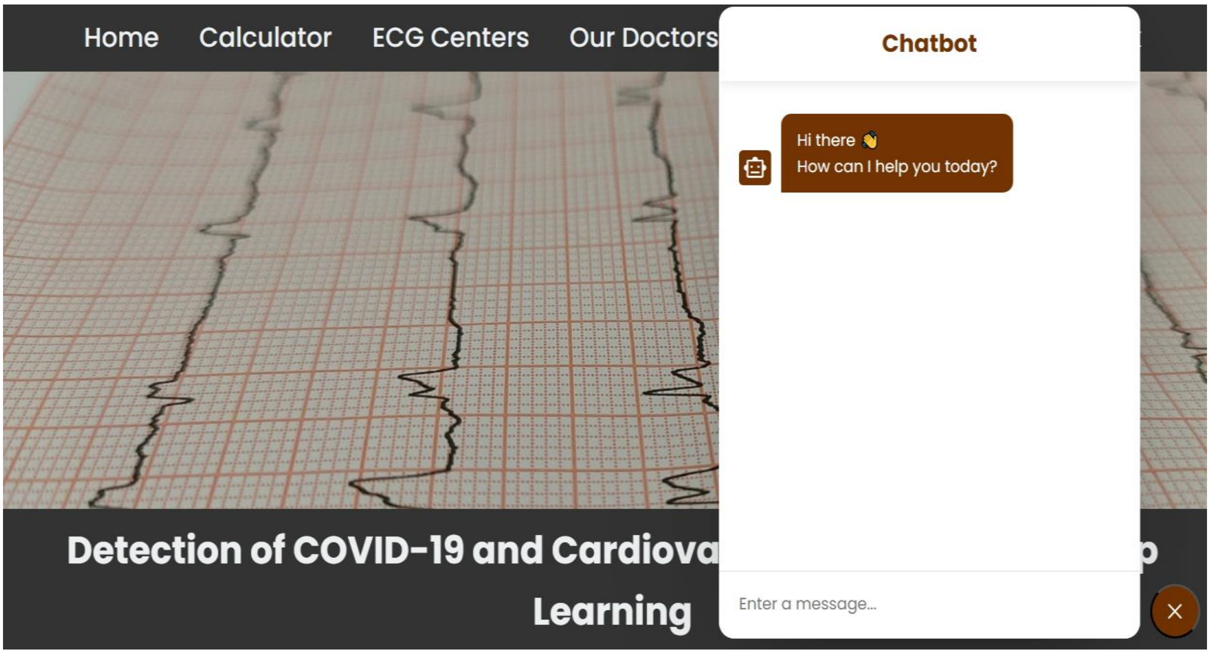


figure (27)

-Patients can initiate a conversation with the Chatbot through the chat interface available on the website.

-The Chatbot responds to patient messages immediately, providing an interactive and instant experience.

1.Answering Inquiries:

-Patients can ask any questions or make inquiries they have about medical services, medications, treatment procedures, and more.

2.Providing Assistance and Guidance.

3.Enhancing the Patient Experience:

-The Chatbot contributes to improving the patient experience by providing instant support and comprehensive answers to their inquiries.

-This helps increase patient satisfaction and confidence in the medical services provided.

7.1.9 Our Doctor Page:

includes:

-Photos of the doctors

-Their contact information, such as clinic addresses and phone numbers

-The ability to communicate with them through social media platforms like WhatsApp, Twitter, and Facebook

This allows patients to not only see the doctors' profiles and contact details, but also connect with them directly through popular social media channels. This can facilitate more personal communication and interactions between patients and physicians beyond just the clinic setting.

The ability to directly book an appointment with the preferred doctor through the website. In this way, the "Doctors" page helps patients find the best qualified and suitable doctors for their healthcare needs.

figure (28)

7.1.10 Patient Feedback Page:

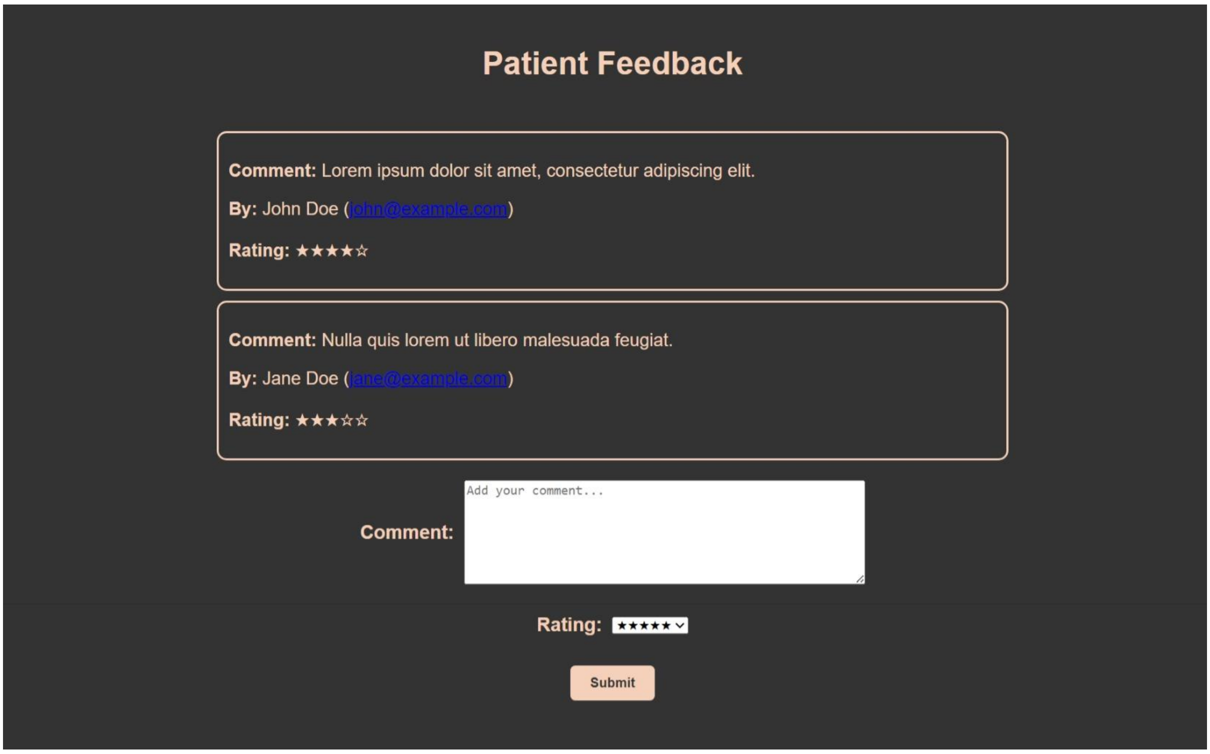


figure (29)

The "Feedback" page on a website is a dedicated page that allows patient to provide their opinions and comments about the site or its services. This page serves as an important communication channel between the site administration and its visitors, enabling patients to share their experiences, suggest improvements, or report issues they encountered. Typically, this page includes a form to fill out with necessary information such as:

Name (optional in some cases)

Email (for follow-up if needed)

rating system, where patients can rate specific aspects of the website or the provided services. These aspects typically include:

Ease of use, Website design, Content quality, Overall satisfaction with the services, Feedback text (the space where the user writes their comment or opinion).

The page may also include options to rate specific aspects of the site (such as ease of use, site design, user satisfaction with the services, etc.) or links to additional help resources.

7.1.11 ECG Centers Page:

This page provides patients with the most important sites where they can perform an electrocardiogram (ECG).

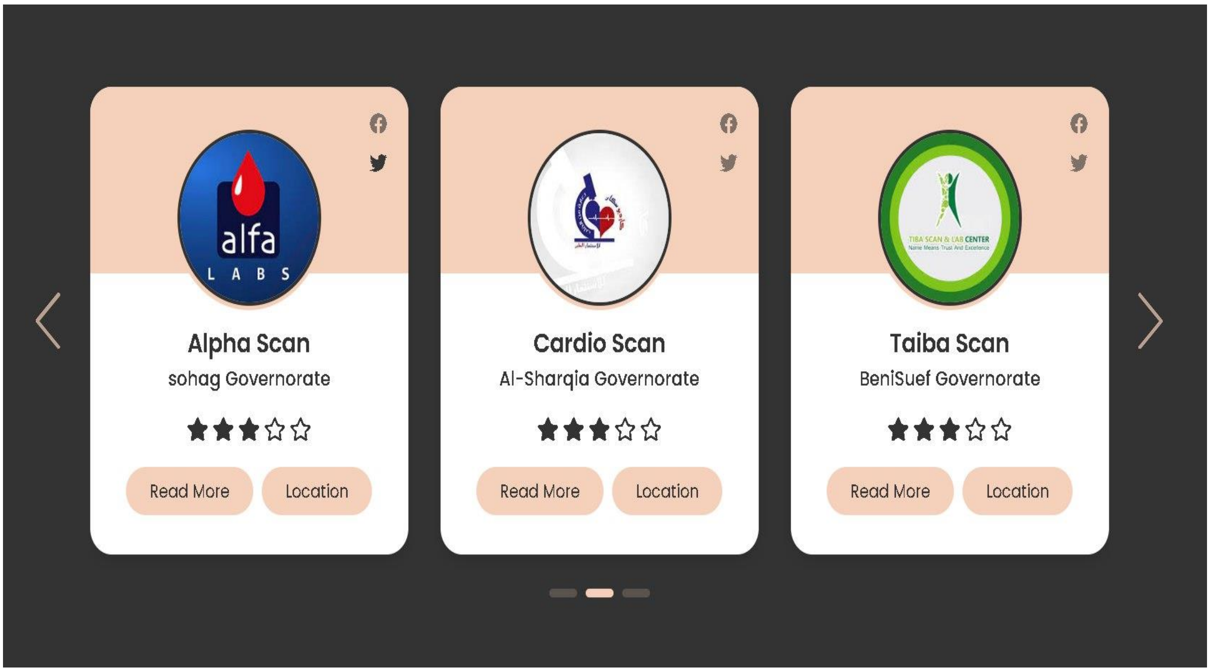


figure (30)

7.1.12 Lifestyle Page:

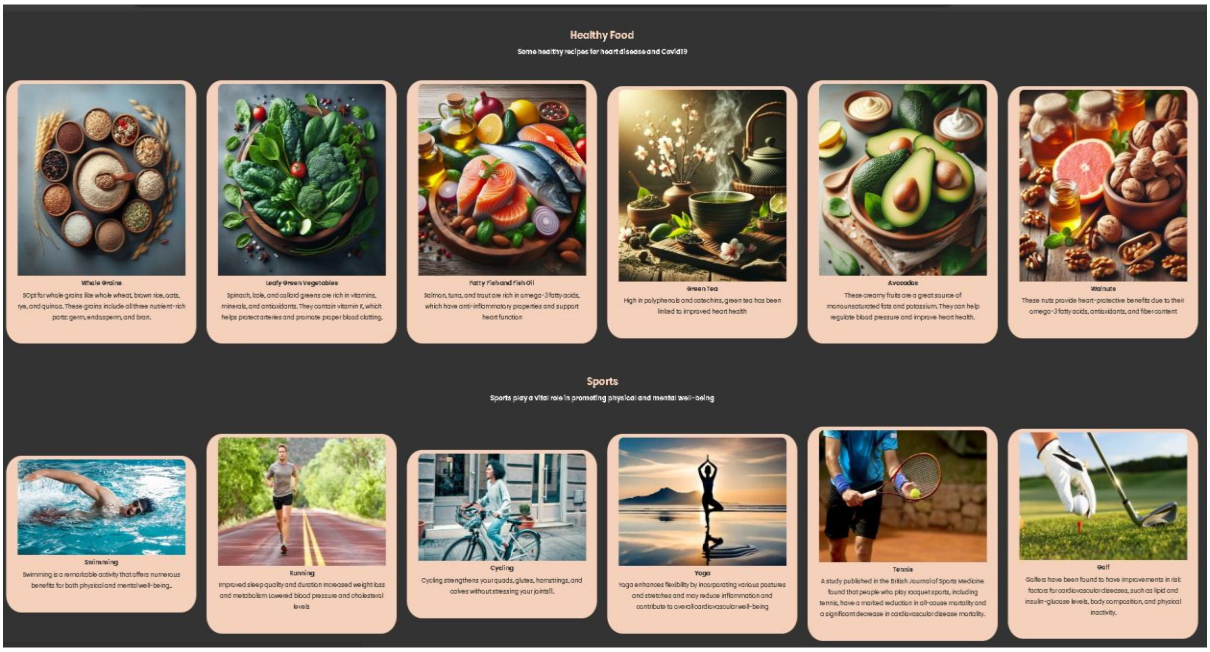


figure (31)

1.Explain the importance of healthy eating and exercise in human life.

-Benefits of following a balanced diet and regular exercise.

2.Healthy Eating Sections:

-Explanation of basic food groups (carbohydrates, proteins, fats, vitamins, minerals, and water).

-Tips for choosing healthy foods (like vegetables and fruits, plant and animal proteins, whole grains).

-Examples of balanced meals.

3.Benefits of Exercise:

-Impact of exercise on physical and mental health.

-Different types of sports and their benefits (aerobic exercises, strength training, flexibility exercises).

7.2 Backend

7.2.1 Authentication (signup, sign in & verification & forget password)

Overview

This module handles the user authentication processes including user registration (signup), login (sign in), email verification, and password recovery (forget password) using Django as the backend framework. The authentication system ensures secure access to the application by implementing industry-standard practices.

Signup (User Registration)

The signup process allows new users to create an account in the system. Users must provide a unique username, a valid email address, and a password. After registration, a verification email is sent to the provided email address. This step ensures that the email address is valid and belongs to the user. The user must verify their email before gaining full access to the system.

Email Verification

Upon receiving the signup request, the system sends a verification email containing a unique token to the user's email address. The user needs to click on the verification link to confirm their email address. This process prevents spam accounts and ensures that the registered email is active and correct. If the token is valid and not expired, the email is marked as verified.

Sign In (User Login)

The sign-in process allows registered users to log into the system using their username and password. Upon successful authentication, the system generates a session token to manage the user's session securely. This token is used for subsequent requests to verify the user's identity and maintain the session state.

Forgot Password

If a user forgets their password, they can initiate the password recovery process by providing their registered email address. The system sends a password reset link to the provided email. This link contains a unique token that allows the user to reset their password. This ensures that only the rightful owner of the email address can reset the password.

Reset Password

Using the password reset link sent to their email, the user can reset their password. The link directs the user to a secure page where they can enter a new password. The system validates the token to ensure it is valid and not expired before allowing the password change. This process helps users regain access to their accounts securely.

7.2.2 Risk Calculator

Overview

This section describes the backend functionality of the Risk Calculator, a tool designed to assess a user's risk of developing diseases, particularly cardiovascular diseases, based on their health and lifestyle data. The backend system processes user input to provide a personalized risk assessment, which helps users make informed health decision.

User Input Data

The Risk Calculator requires the following data from the user to evaluate their risk:
1. Age (years): The user’s age, as age is a significant risk factor for many diseases.

2. Total Cholesterol (mg/dL): The total cholesterol level in the user’s blood, which is a key indicator of heart disease risk.

3.Systolic Blood Pressure (mmHg): The user’s systolic blood pressure, indicating the force exerted by blood on artery walls during heartbeats**.**

4. HDL Cholesterol (mg/dL): The level of high-density lipoprotein (HDL) cholesterol, known as “good” cholesterol, which helps reduce heart disease risk.

5. Physical Activity (hours per week): The amount of time the user spends on physical activities each week, influencing overall health.

6. Alcohol Consumption (drinks per week): The number of alcoholic drinks the user consumes weekly, as excessive alcohol can increase health risks.

7. Weight (kg): The user’s weight, used to calculate body mass index (BMI) and assess overall health.
8.Height (cm): The user’s height, also used in calculating BMI.

9.Smoker: Whether the user smokes, as smoking is a major risk factor for many diseases.

10.Diabetes: Whether the user has diabetes, which significantly raises the risk of cardiovascular diseases.

11-Family History of Heart Disease: Whether the user has a family history of heart disease, as genetics can influence disease risk.

Data Collection

The backend system collects the user's data through a secure interface. Users input their health and lifestyle information, which is then stored in the system for processing and forThe doctor can examine these results in the doctor's section of the project.

The primary goals of the Risk Calculator are to:

- **Inform:** Provide users with a clear understanding of their health status and risk factors.
- **Educate:** Increase awareness about the impact of various health metrics on disease risk.
- **Motivate:** Encourage users to adopt healthier lifestyles based on their personalized risk assessment.
- **Guide:** Assist users in making informed decisions about seeking medical advice or interventions to manage their health.

7.2.3 Upload Image

Overview

This section describes the process by which users can upload an electrocardiogram (ECG) image for health assessment. The uploaded image is stored securely and analyzed by an artificial intelligence (AI) model to determine whether the user is healthy or at risk of heart disease. Users can also update the image later to confirm or reassess the AI's analysis.

ECG Image Upload

Users can upload an ECG image through the system's interface. This image is essential for the AI model to analyze heart health.

1. **Image Upload Interface:** The system provides a user-friendly interface for uploading ECG images. Users can select the ECG image file from their device and submit it securely.
2. **Storage:** The uploaded ECG image is stored securely in the user's profile. The system ensures that each image is correctly associated with the respective user while maintaining data privacy and security.

AI Analysis

The uploaded ECG image is processed by an AI model trained to detect signs of heart disease. The analysis involves several steps:

1. **Preprocessing:** The image undergoes preprocessing to ensure it is in the correct format and quality for analysis. This includes operations like resizing and normalization.
2. **AI Model Processing:** The preprocessed ECG image is analyzed by the AI model. The model, trained on a large dataset of ECG images, detects indicators of heart health or disease.
3. **Result Generation:** Based on the analysis, the AI model generates a result indicating whether the user appears healthy or shows signs of heart disease. This result is communicated back to the user.

User Notification

Upon completing the AI analysis, the user is notified of the results. Notifications can be accessed through the user interface .

1. **Healthy:** If no signs of heart disease are detected, the result will indicate that the user appears to be healthy and print 'Normal Person.
2. **At Risk:** If potential signs of heart disease are detected, the result will indicate that the user may be at risk and should seek further medical evaluation as print Any of these specific diseases will appear:

   'Myocardial Infarction Patient', 'Patient that have abnormal heartbeats', 'COVID-19 Patients', 'Patient that have History of MI'.

Image Update

Users can upload a new ECG image at any time. This feature allows users to:

1. **Reassess Health Status:** Users can confirm the AI's previous analysis or check for changes in their heart health**.**
2. **Confirm Results:** Updating the image helps users verify or refute the initial findings, especially if they have taken steps to improve their heart health.

7.2.4 User profile

The Profile of User page consolidates all relevant data associated with a user, including login information, risk calculator entries, and uploaded ECG images. This page is also accessible to doctors for medical examination and review.

Contents of the Profile Page:

1. **User Data from Register:**
   - **name:** The unique identifier chosen by the user.
   - **Email Address:**The user's registered email for communication and notifications.
   - **Contact Information**: Phone number or other contact details provided by the user.
   -
2. **Risk Calculator Entry:**
   - **Age:** The user’s age, which is a critical factor in risk assessment.
   - **Total Cholesterol (mg/dL):** The user’s cholesterol levels used to evaluate cardiovascular risk.
   - **Systolic Blood Pressure (mmHg):** The user’s blood pressure reading.
   - **HDL Cholesterol (mg/dL):** Levels of good cholesterol in the user’s blood.
   - **Physical Activity (hours per week):** The user’s weekly physical activity.
   - **Alcohol Consumption (drinks per week):** The amount of alcohol consumed by the user per week
   - .**Weight (kg):** The user’s weight.
   - **Height (cm):** The user’s height.
   - **Smoking Status:** Whether the user smokes.
   - **Diabetes Status:** Whether the user has diabetes.
   - **Family History of Heart Disease:** Presence of heart disease in the user’s family history.
   - **Risk Score and Level:** The calculated risk score and corresponding risk level.

3. **Electrocardiogram (ECG) Image:**
    - **Uploaded ECG Image:** The ECG image uploaded by the user for analysis.
    - **Analysis Results:** The results of the AI analysis indicating heart health status.
    - **Option to Update Image:** Feature allowing users to upload a new
    - ECG image for re-assessment.

Accessibility for Doctors:

- **Doctor's Access:** Doctors can view the user's profile page to examine the provided data.
- **Review and Recommendations:** Doctors can review the risk calculator entries and ECG analysis results to provide medical advice.
- **Data Consistency:** Ensures that all relevant health information is available in one place for comprehensive evaluation.

7.2.5 Add Feedback

The Add Feedback section allows users to share their opinions about the site and view comments from other users. User feedback, including star ratings, is recorded and saved along with their name and email.

Functionality:

1. **Submitting Feedback:**
    - **Feedback Form:** Users can access a form to submit their feedback.
    - **Text Input:** Users can write comments detailing their experience and opinions about the site.
    - **Star Rating:** Users can rate their experience on a scale (e.g., 1 to 5 stars).
    - **Submission:** The form includes a button to submit the feedback.
2. **Recording Feedback:**
    - **User Identification:** Feedback is recorded along with the user's name and email.
    - **Data Storage:** All feedback data is securely stored in the database, associated with the user's profile.

3. **Viewing Feedback:**
    - **Comments Section:** Users can view a section displaying feedback from other users.

7.2.6 Doctor Section

In this section, doctors who register on the site gain access to a user interface tailored to their needs. After registration, they can view user data in the Registered Page field. Here, they can access user-entered data and provide comments or diagnoses, which will be sent to the user's(patient) personal page. Below are the key points

User Interface for Doctors:

1. **Registered Page Field:**
    - Upon successful registration and login, doctors are directed to the Registered Page field.
    - This section displays a list of registered users(patients) whose data the doctor can access and review.
2. **View User Data:**
    - Doctors can view user-entered data, including medical history, risk calculator entries, and uploaded ECG images.
    - User data is presented in an organized format for easy review and analysis.
3. **Comment or Diagnosis:**
    - Doctors have the ability to write comments or diagnoses based on their assessment of the user's condition.
    - They can provide personalized feedback or recommendations to the user.
    - Users should be aware of any feedback or recommendations provided by healthcare doctors by visiting their page.
4. **Communication with Users:**
    - Comments or diagnoses written by doctors are sent directly to the user's personal page.
    - This facilitates communication between doctors and users, allowing for personalized interaction and feedback.

7.2.7 Connect the model

1.Setting Up the Flask Environment

- To begin, I ensured Flask was installed in my development environment. Flask, a lightweight web application framework in Python, is ideal for creating web applications and APIs.

app.py ×

app.py > ...

```python
from flask import Flask, request, jsonify, render_template
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array, load_img
import numpy as np
```

Flask: A micro web framework for Python to handle web requests and responses.

load_model: A function from TensorFlow Keras to load a pre-trained model.

img_to_array, load_img: Utility functions from TensorFlow Keras to preprocess images.

numpy: A library for numerical operations, used here for handling image data arrays.

Initializing Flask App

```python
app = Flask(__name__)
```

2.Loading the Deep Learning Model

- I confirmed that all necessary libraries, particularly TensorFlow and Keras, were installed. These libraries are crucial for loading and using deep learning models.
- Within the Flask application, I successfully loaded the pre-trained model, specifically ‘Train_model5_inceptionv3old.h5’..This involved initializing the model in the application’s runtime environment.
- Loads the pre-trained InceptionV3 model for ECG image classification from the specified path.

```
model = load_model('trained_model5_inceptionv3old.h5')
```

Defining the Prediction Route

```
# Define a route to handle image upload and prediction
@app.route('/predict', methods=['POST'])
def predict():
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})
    file = request.files['file']
    if file.filename == '':
        return jsonify({'error': 'No selected file'})
    if file:
        # Load and preprocess the image
        img = load_img(file, target_size=IMAGE_SIZE)
        img_array = img_to_array(img)
        img_array = np.expand_dims(img_array, axis=0) / 255.0

        # Make prediction
        predictions = model.predict(img_array)
        predicted_class = np.argmax(predictions, axis=1)[0]

        # Map the predicted class to the category
        categories = ['history_of_microcardial_ecg', 'microcardial_ecg', 'normal_ecg', 'abnormal_ecg', 'cov
        result = categories[predicted_class]

        return jsonify({'prediction': result})
```

@app. route ('/predict', methods=['POST']): Defines the `/predict` endpoint that accepts POST requests.

request.files['file']: Retrieves the uploaded file from the request.

file.filename: Checks if a file is selected.

load_img(file, target_size=IMAGE_SIZE): Loads the uploaded image and resizes it to the target size.

img_to_array(img): Converts the image to a numpy array.

np.expand_dims(img_array, axis=0) / 255.0: Expands the dimensions of the image array to match the model's input shape and normalizes the pixel values.

model.predict(img_array): Makes predictions using the loaded model.

np.argmax(predictions, axis=1)[0]: Gets the index of the highest predicted probability, indicating the predicted class.

categories: A list of class names corresponding to the model's output classes.

result = categories[predicted_class]: Maps the predicted class index to the corresponding category name.

return jsonify({'prediction': result}): Returns the prediction result as a JSON response.

Running the Application

```
if __name__ == '__main__':
    app.run(debug=True)
```

3.Creating an Endpoint for Predictions

- An endpoint was established to handle prediction requests. This endpoint accepts input data from users, processes the data through the model, and returns predictions.
- It was essential to ensure that the input data underwent appropriate preprocessing, mirroring the preprocessing steps applied during the model's training phase. Similarly, the output from the model was formatted to be user-friendly and easily interpretable.

4.Running the Flask Application Locally

- I launched the Flask server on my local machine to test the application's functionality. This step was crucial for identifying and resolving any issues before deployment.
- Using tools like Postman, I sent various test requests to the prediction endpoint, verifying that the responses were accurate and met the expected criteria.

Deploying the Flask Application

5.Deploying on Render

- Render Account Setup: Created an account on Render, a platform that simplifies the deployment of web applications.
- Repository Linking: Linked my GitHub repository containing the Flask application to Render. This allows Render to access the codebase directly.
- Service Creation: Created a new web service on Render. During this setup, specify the build command and start command that Render uses to deploy the application.
- Environment Variables: Configure any necessary environment variables in Render's dashboard to ensure the application runs smoothly.

6.Linking to the Site

- API Integration: Integrated the API with the website's frontend using JavaScript and AJAX. This integration enables seamless communication between the client-side and server-side components, allowing users to submit and receive predictions dynamically.
- CORS Handling: Configured Cross-Origin Resource Sharing (CORS) to handle scenarios where the frontend and backend are hosted on different domains. This ensures secure and permissible data exchange between the domains.

Chapter 08

Conclusion

8.1 Conclusion:

Our website provides comprehensive services for diagnosing heart-related diseases and COVID-19 using ECG images. Here's what you can upload your ECG images and receive a diagnosis for potential heart-related conditions and utilize our platform to detect possible signs of COVID-19 from your ECG.

Additional Medical Services: Locate nearby imaging centers where you can perform necessary scans and tests and access a directory of specialized doctors in cardiac diagnostics and connect with them for consultations.

Patients can leave comments and questions about their diagnosis, and doctors can respond directly on the platform. Doctors can also initiate discussions and provide feedback.

Our platform aims to facilitate easy access to heart disease and COVID-19 diagnostics, while also providing a robust support system through direct interaction with medical professionals and access to essential health services

8.2 Future scope:

. Artificial Intelligence and Deep Learning: Utilizing AI and deep learning technologies to enhance diagnostic accuracy and speed. Over time, models can become more precise and efficient in analyzing ECG images and recognizing complex patterns.

Database Expansion: Increasing the volume and quality of data used to train AI models can improve system performance, making it more reliable and accurate.

Electronic Health Record (EHR) Integration: Integrating the website with EHR systems to facilitate easier and more effective access to ECG diagnostics for doctors and nurses.

Contribution to Medical Research: The website can contribute to cardiac disease research by providing accurate data and analyses, aiding in the development of new treatments and a better understanding of heart diseases.

Mobile App Development: Developing a mobile app to allow patients and healthcare providers to access ECG analysis results on the go.

Data Sharing and Collaboration: Facilitating data sharing and collaboration among healthcare providers, researchers, and institutions to improve collective understanding and treatment of cardiac conditions.

8.3 TimePlane:

Research and understand the idea of the project	2 weeks
Understanding the dataset	2 weeks
Data visualization	2 weeks
Preprocessing for data and augmentation ,Create a website and create database in MySQL	4 weeks
Implementation and build the model. Link model and make Api for website.	4 weeks
-Improve accuracy ,add new features to website	4 weeks
Obtain the final version of the project and to amend any proposals from the doctor and teacher assistant.	2 weeks

8.4 Reference:

[1] Y. Fan, K. Zhao, Z.-L. Shi, et al., Bat coronaviruses in China, Viruses 11 (2019) 210–223.
[2] S. Chauhan, Comprehensive review of coronavirus disease 2019 (COVID-19), Biomed. J. 43 (2020) 334–340.
[3] B. Abhijit, et al., A deep learning based approach for automatic detection of COVID-19 cases using chest X-ray images, Biomed. Signal Process Control 71 (Part B) (2022) 103182.
[4] D. Dong, Z. Tang, S. Wang, et al., The role of imaging in the detection and management of COVID-19: a review, IEEE Rev. Biomed. Eng. 14 (2020) 16–29.
[5] V.M. Corman, O. Landt, M. Kaiser, et al., Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR, Euro Surveill. 25 (2020) 23–30.
[6] T. Ai, Z. Yang, H. Hou, et al., Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases, Radiology 296 (2020) E32–E40.
[7] T. Iwasawa, M. Sato, T. Yamaya, et al., Ultra-high-resolution computed tomography can demonstrate alveolar collapse in novel coronavirus (COVID-19) pneumonia, Jpn. J. Radiol. 38 (2020) 394–398.
[8] L.A. Rousan, E. Elobeid, M. Karrar, et al., Chest x-ray findings and temporal lung changes in patients with COVID-19 pneumonia, BMC Pulm. Med. 20 (2020) 1–9.
[9] O. Attallah, X. Ma, Bayesian neural network approach for determining the risk of re-intervention after endovascular aortic aneurysm repair, Proc. IME H J. Eng. Med. 228 (2014) 857–866.
[10] A. Karthikesalingam, O. Attallah, X. Ma, et al., An artificial neural network stratifies the risks of Reintervention and mortality after endovascular aneurysm repair; a retrospective observational study, PLoS One 10 (2015), e0129024.
[11] O. Attallah, MB-AI-His: histopathological diagnosis of pediatric medulloblastoma and its subtypes via AI, Diagnostics 11 (2021) 359–384.
[12] M.A. Ozdemir, G.D. Ozdemir, O. Guren, Classification of COVID-19 electrocardiograms by using hexaxial feature mapping and deep learning, BMC Med. Inf. Decis. Making 21 (2021) 1–20.
[13] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (2019) 1–48
. [14] S.U. Amin, M. Alsulaiman, G. Muhammad, et al., Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion, Future Generat. Comput. Syst. 101 (2019) 542–554.
[15] Q. Xu, Z. Wang, F. Wang, et al., Multi-feature fusion CNNs for Drosophila embryo of interest detection, Phys. Stat. Mech. Appl. 531 (2019), 121808
. [16] Q. Zhang, H. Li, Z. Sun, et al., Deep feature fusion for Iris and periocular biometrics on mobile devices, IEEE Trans. Inf. Forensics Secur. 13 (2018) 2897–2912.
[17] L. Hongtao, Z. Qinchuan, Applications of deep convolutional neural network in computer vision, J. Data Acquis. Process. 31 (2016) 1–17.